DIAMM database documentation

April 2011

Julia Craig-McFeely

1 HISTORY
1.1 Software 1
1.2 Data 2
1.3 Deficiencies and workarounds 2
1.4 Sharing data with MySQL 3
1.5 Bugs 5
1.6 Evolution and Development 6

2 CONTENT
2.1 Layouts 16
2.2 Data 16
2.3 Tables and Fields 17
2.4 Entity Relationship Diagram (ERD) 49

3 USER INFORMATION
3.1 Editing existing records 50
3.2 Creating new records 51

1. HISTORY

1.1 Software

The DIAMM database was originally constructed and populated in Filemaker Pro version 4.0. This version of Filemaker (henceforward 'FM'), although claiming to be relational, actually created data tables in individual files, and these files were linked via relationship fields.

Version 7.0 however changed the software drastically, making it truly relational, so all tables of a database now exist within a single 'file' or 'document'.

Filemaker is a very forgiving medium, allowing the user to create a custom workspace that many other softwares would not allow. The GUI allows the creation of custom layouts to suit numerous different types of task, facilitating quick and easy data input for all types of job.

The software is also cross-platform (Mac/PC) which was a major reason for its selection early on in the project, since participants used both platforms.

Primary platform: Mac

1.2 Data

The core data was originally created to allow the team to control information about libaries that it was visiting to undertake photographic work: it held contact information, summaries of corresponence and licensing agreements, and lists of manuscripts, basic descriptions of the material to be photographed, and once the photographs were taken, a list of image filenames and the pages to which they referred. It was not originally conceived as the back-end to an online delivery system.

The database grew organically with need, and was therefore not planned from the outset at the level of complexity that it now shows. The original core tables were:
Archive (which included country and city information, now migrated to separate authority lists, and information about licensing and correspondence, dates of visits and other information relating to imaging in that library or collection)
Source (related to Archive, which contained only very basic physical information such as page-extent and measurements, to give the photographer information to allow for planning of imaging work)
Image (related to Source, and containing information about the images taken at the archive)
Photographer (information about the photographer who created the images - this table is no longer included in the data structure)

The first change to the content was necessitated by the growth in the number of archives and countries being visited: originally the dataset was only designed for UK libraries, and so there was no need to indicate the country. When the project extended into Europe in its second phase **country** and **city** tables were added and the data was adjusted to utilise these new tables.

Shortly after this the project received funding to deliver the image content online, and it was clear that it would be necessary to accompany the images with a metadata framework in which to nest the images so that they could be viewed not simply as isolated images, but components of a clearly described object. It made sense to utilise an extended version of the existing database, and this led to the first major extension to content. The descriptions were available in two printed catalogues (RISM and the Census Catalogue), both well known to scholars of the period. Permission was obtained to put the catalogue entries in the RISM volumes into the database, and the **Source** table was expanded considerably to allow this inclusion.

The catalogue information contained lengthy bibliographies, expressed in the printed books in the form of a list of abbreviations which referred to a master bibliography appended to the volumes. Rather than simply reproduce these relatively meaningless lists a bibliography table was added and related to the **Source** table via an intersection set and using the bibliographical abbreviation to connect the tables.

Funding allowed the creation of an **Item** table to represent musical compositions within each source. This was the first table that fell outside the simple top-down hierarchical model that had been used hitherto and, since many items could appear on a page (which was the same as an image) and many pages could comprise an item it was at the same level as the **Image** table, but connected to that via intersection set.

A data merge between DIAMM and the Motet database of Thomas Schmidt-Beste led to the creation of a full-text table (**Text**) and its associated satellite tables (language, etc.), and led to a formalisation of the Item information involving the creation of numerous intersection sets and authority lists, including relating items, **composers**, composition models, individual **compositions** etc. to the **Bibliography** table. The **Bibliography** table was also re-engineered (and had an '**Author**' authority list added via an intersection set) to allow the bibliographical items to be delivered using a house style.

1.3 Deficiencies and workarounds

A major defect in Filemaker by comparison with e.g. MySQL is that relationships between data in tables at more than one remove are not reliable. For instance if the user wishes to see all images in a city (regardless of the archive), the data would need to be mined via the following set of relationships:

City→Archive→Source (i.e. manuscript)→Image

Thus requiring navigation through three tables. Because all these tables are relatedto each other the software should return a list of images related only to the master table (in this case the City table), but in FM it does not (or at least not reliably). In MySQL the data delivered seems to correctly contain only the data set belonging to the city concerned, but in FM the data delivered tends to be scrambled, so the image list delivered is not reliable. This means that in order to mine data in the Image table from the City table, it is necessary to 'short-circuit' the relationship manifest above, by creating a link directly between the City and Image tables. Since FM will refuse to make a relationship between tables where one already exists, it is necessary to create a secondary relationship, bypassing the intervening tables. In the ERD (Entity Relationship Diagram) this creates a proliferation of tables, and in the actual data it involves the creation of some otherwise redundant fields:

City has a unique identifier cityKey field
Archive connects to City using the cityKey data. Archive has a unique identifier archiveKey field
Source connects to Archive using the archiveKey data. Source has a unique identifier sourceKey field.
Image connects to Source using the sourceKey data and has it's own unique identifier imageKey field.

In order to bypass the interim relationships in the Archive and Source tables, the Image table has to contain the cityKey data, allowing the user to create a direct relationship between Image and City via the cityKey fields in each table. This is long-winded and should be unnecessary, but has been used as a workaround when creating layouts that display data related to the master record being viewed (often essential when dealing with complex information). This example is only one of many, which means that the working database that is used for populating content has an untenably complex ERD, and many 'redundant' fields in each table in order to facilitate these relationships.

In addition to these redundant structures, work of the data team was supported by including access to the image content—both images of the catalogues and images of manuscripts—in layouts. This allowed e.g. a technician proof-reading OCR from a printed catalogue to see the relevant page-image from the catalogue in the same layout as the text being proofed.

Rather than contain these images in the master databsae (which would make it extremely large and unwieldy), two external datasets were created (both around 6-8 GB in size), one containing catalogue images and one manuscript images, and the primary textual dataset was connected to them via external relationship. (It is worth noting at this point that PC files have a size limit of 4GB, which means that both the image databases in constant use for DIAMM would not have been possible to manage on a PC platform. This (together with the better graphics management of Mac vs PC) was one reason for the choice of the Mac platform at the project outset (1998).)

In order to manage the expression of the dataset outside the project therefore, a cut-down version of the database was created, known as **DIAMM_Shell**, in which the redundant relationships and fields were eliminated, creating an intelligible ERD which, although still complex, is sufficiently concise that it can be understood by a database technician. This is the ERD in use in this document.

1.4 Sharing data with MySQL

The web delivery of data is from a SQL database, which means that all FM content must be exported from FM and imported to SQL. Compared to SQL, FM is extremely forgiving, allowing redundant relationships (see above) and various other character and content oddities which blew out the SQL version when the data was exchanged.

In earlier versions of FM exchanging data caused some considerable problems, since ODBC support in FM—even in versions 7-9—was shaky, to say the least. Exchange was managed through perl scripts that first exported

the data from FM, then imported it to SQL, a tedious and time-consuming process, which had its own problems caused by e.g. default field character limits in SQL. Later versions of Filemaker improved communication radically (version 10), and not only allowed a Filemaker front end to display data from a SQL back end, but also to import/export data seamlessly between the two platforms. This however requires that the SQL end to be set up in a very specific way:

Since FM allows certain types of data that SQL does not (e.g. empty content in fields defined as numeric; no default character limit on field lengths) and does not recognise certain types of data (e.g. there is no such thing as a 'null value' in FM), the SQL end has to be set up so that fields allow the FM content (i.e. no character limit).

Unfortunately one problem remained in version 10, that of export character sets (i.e. Unicode export) which quickly manifested itself in accents common to e.g. German and Polish place names and texts. FM version 11 fixed this problem, but because version 11 has some significant bugs most data input is still done in v. 10.

Although FM allows the user to format field content (e.g. bold, italic, fonts, colours etc.) that formatting is not carried over when exporting to SQL, since SQL does not recognise formatting.

Field lengths in the DIAMM dataset vary massively from single-character Y/N or numeric fields to fields which contain a discursive text description of a document that can run to many paragraphs and thousands of characters. This is because much of the DIAMM data was created outside a database environment (e.g. printed catalogue descriptions of documents) and never intended for a database environment. These field lengths and content aspects can be a problem when communicating with SQL, which is designed for smaller content parcels (e.g. it has a default field-length setting for new fields of 256 characters).

The most difficult problem that arises when exporting FM tables to SQL is that FM allows 'broken dependencies'. In other words, if we create a source record and link bibliographical items and inventory items to it via an intersection set or direct relationship, then we delete the source record, the dependency records in the other tables remain, and do not cause the FM database to choke. If we then run an upload of these tables to SQL, the SQL database will not accept records with orphan dependencies, so the upload registers failed records or fields. In order to fix this, a manual integrity check has to be run on all intersection sets and dependent tables before attempting an upload, and the orphan records deleted. In fact, the failure of these records to upload is not a problem, since they are no longer required, but if the upload returns a failure message we need to know whether there is a rogue record that has to be chased down, so it is best to delete the orphans and then if there is an upload error we know that there is a real fault in the data.

It is possible to upload single tables in the database as 'updates' to the existing content: so the upload system looks for records in the table that match the key number of the source database, and rewrites the data in those records. This is a quick way of registering content updates to existing records, but will not work if new records have been created unless an update is run on ALL tables in the correct order. The safest way to deal with new records is to delete the content and upload every table in the correct sequence so that failures are not caused by orphan dependencies. Tables cannot be deleted if other tables relate to them, so the content delete has to be done in the correct sequence as well. The list below gives the correct upoad sequence. The delete sequence follows this list in reverse. The uploads have all been scripted in the upload database, in the correct sequence, as has the delete sequence.

FM table name	equivalent SQL table name
alCountry	alCountry
alCity	alCity
alProvenance	alProvenance
alAuthor	alAuthor
alCopyist	alCopyist
alCopyistType	alCopyistType

alSettype	alSettype
alAffiliation	alAffiliation
alPersonRelationship	AlPersonRelationship
alPerson	alPerson
alGenre	alGenre
alVoice	alVoice
alClef	alClef
alLanguage	alLanguage
alNotationType	alNotationType
AlMensuration	alMensuration
Archive	Archive
CatalogueIndexes	CatalogueIndex
Set	Set
Bibliography	Bibliography
Composition	Composition
Source	Source
Item	Item
Image	Image
SecondaryImages	SecondaryImage
Composers	Composer
Text	Text
SourceProvenance_IS	SourceProvenance
AuthorBibliography_IS	AuthorBibliography
BibliographyPage_IS	BibliographySource
Source_Copyist_IS	SourceCopyist
SourceSet_IS	SourceSet
SourcePerson_IS	SourceAlPerson
BiblItem_IS	BibliographyItem
BiblComposer_IS	BibliographyComposer
CompositionGenre_IS	CompositionGenre
BiblComposition_IS	BibliographyComposition
Composerlink_IS	CompositionComposer
alCycleType	alCycleType
Composition_cycle	CompositionCycle
Cycle_Item_IS	CompositionCycleComposition
alLanguage	alLanguage
TextLanguage_IS	TextLanguage
BiblText_IS	BibliographyText
ItemImage_IS	ItemImage

1.5 Bugs

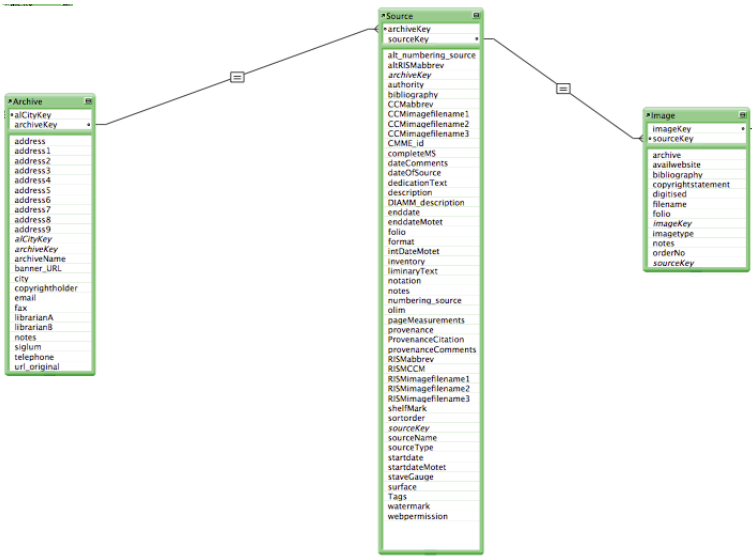
When uploading to the SQL database the server will crash if the upload returns an error message IF copies of any databases on the server are open on any computer other than the one where the upload is being managed. This also (unfortunately) happens if another database is open on the server which has a connection to the same ODBC data source. An attempt to replace some fields in FM layouts with their SQL counterparts had to be abandoned, since this caused the Filemaker server to crash when an upload to the SQL database was done. It also apparently caused entire table uploads to fail.

FM v.1 1 Layout Menu system: if folders or dividers are introduced the menu fails to display the complete or partial list of layouts either in browse or layout view, and hides 'foldered' sets of layouts, even when they are removed from the folders. The effect of this problem took some time to fix.

1.6 Evolution and Development

The DIAMM database evolved over a very long period (approx 12 years) and changed in content to reflect both the changing needs of the users of the data, and the changing purpose for which the database is used. The following diagrams show original tables for the first DIAMM web app in green, new master tables in red, authority lists in blue and intersection sets in yellow. The full ERD (slightly larger) appears on the last page of this document

Originally, the database was constructed as an efficient way of keeping a list of a relatively small number of manuscripts that were to be photographed for a publication. There were three tables: **Archive**, **Source** and **Image**. The **Archive** table gave details of libraries: addresses, contact names, information about copyright and correspondence regarding a visit from the project photographer, the progress of negotiations etc. The **Source** table simply listed the manuscripts with their dimensions and gave the photographer a description so that they had some idea of what they would be looking at when they got to the library, and how much time to allow for the work. The **Image** table simply listed the image filenames originally, but was expanded to include information about photographers (now omitted). The first ERD then would have looked something like this:

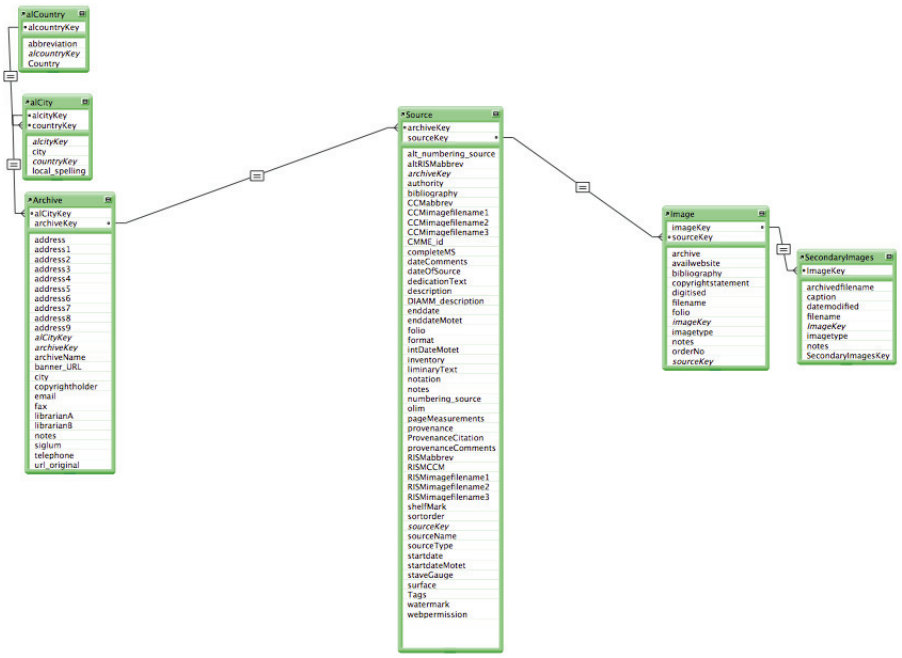


At this point, since all the libraries concerned were in one country, there was no need for **Country** or **City** authority lists. These were added later when the project's imaging activity extended outside the UK. Originally second shots of a leaf were listed as separate images, but this was clearly not a good way to deal with different types of shots of the same page, and as the dataset began to extend to become an information resource about the manuscripts the need for proper management of image information as it related to pages or folios led to the creation of a **Secondary Image** table which listed e.g. UV or restored versions of a page.

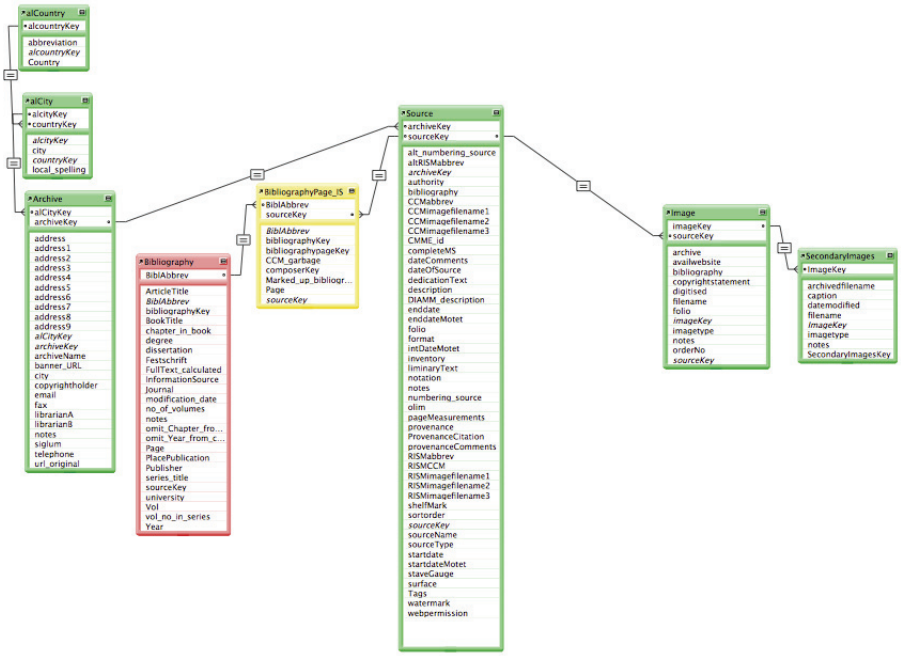
The new structure was sufficient to cope with all the information necessary for managing a collection of images. Fields were added to the **Source** table so that searching was more efficient in a number of key areas such as date, format, size and so on. Around this time the source descriptions were also expanded to include the complete data provided by the printed catalogues *RISM* and *CCM*. Although out of date sometimes by 30-50 years, these catalogues were the primary source of physical and content information about manuscripts, and the detail they provided was extremely useful in cases where a librarian could not easily locate a manuscript in their collection. In some cases the shelfmark had changed, so a facility was necessary to record alterations to shelf marks both for administrative use, and sometimes as the only source of that information available to scholars.

The database was still not being used to form the back-end to an online delivery system, but around this time it also became clear that the quantity of images and the demand for online delivery would need a database management system to handle a web-delivery mechanism. Rather than discarding the existing database, which was populated in considerable detail by this point, the web design team at the then Centre for Computing in the Humanties at King's College London decided to work from the Filemaker database already in use.

A SQL analog was created that mirrored the following structure in order to run the online database delivery system for the existing and expanding image collection. Since the working FM database was constantly being updated this was kept separate from the delivery database, but updates were run at regular intervals when new MSS had to be added to the online collection



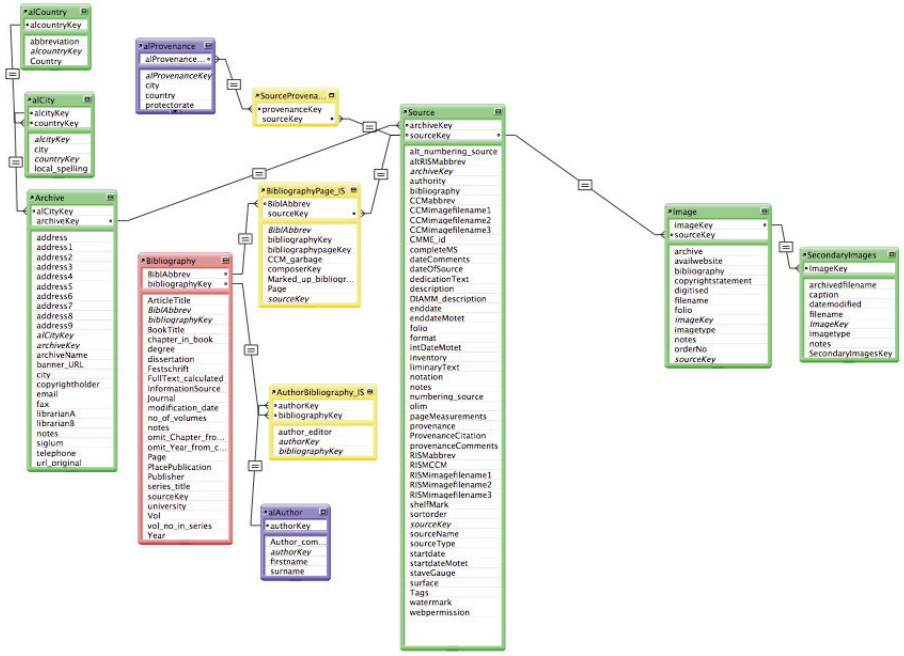
Structural change then began to happen very fast: information input from the printed catalogues included bibliographies for the MSS, but to save print space (since many of these were repetitious) these had been provided as algebraic abbreviations, with a key list giving the full citation. In order to be of better use to end users the database was expanded so that the full bibliographical citation (in the **Bibliography** table) could be keyed to the abbreviation given in each MS listing.



An intersection set allowed for individual page references in each citation and any other notes pertinent to the instance of the reference in the **Source** description.

One of the key search areas online was the provenance field in the **Source** table, and this was inefficiently managed by a repetitive text field. The addition of a simple **provenance** authority list was first implemented, but as the content grew in complexity an intersection set was added to allow more detail about provenance to be added where it was available, without unnecessarily expanding the main **provenance** authority list, which

was limited to countries at that point. Given that the intention was eventually allow users to search on the authors of bibliographical items, an **Author** authority list with intersection set was also added.



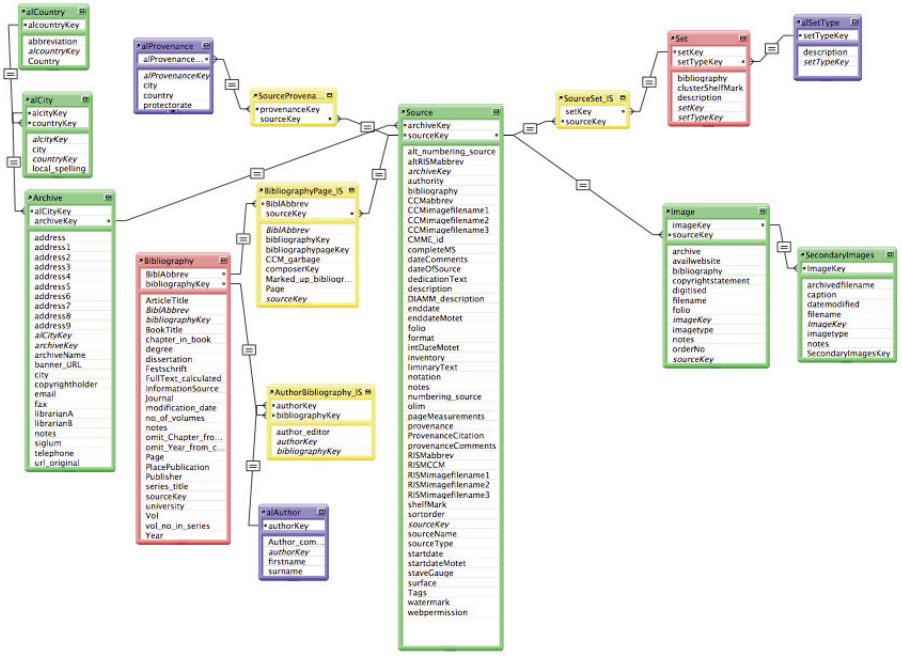
With the expansion in data content, and the increased involvement of a larger number of researchers in creating and using the data, we found that there was a need for a facility that would allow us to group manuscripts together in different types of Sets. Some groups of sources that were spread over many libraries and countries belonged together since they were fragments that had been dispersed over time. This was one type of set. Another type was of members of partbooks – groups of books copied separately and each providing a single voice part for a multi-voice work. Some of these books were shelved together in one library, while others had been dispersed much as the fragments had been. Up to this point we had listed sets of partbooks in a library under one shelfmark as a single 'source'. However it became evident when dealing with descriptions that in some cases each book might require its own description, rather than sharing a blanket description with the whole set. The decision was taken therefore that a each entry in the **Source** table should represent a single book or manuscript item, and that sets should be linked together by causing them to belong to a master item in a '**Set**' authority list. There was (briefly) a question as to whether the **Set** table should act as an intersection set between the **Archive** and **Source** tables, but as this would have meant that the great majority of sources would belong to a set of one, this option was quickly abandoned. As it turned out the necessity for some sources to belong to more than one set necessitated the insertion of an intersection set in any case, between the **Source** and **Set** tables.

A further authority list was required to define the type of set (**alSetType**) that the source belonged to, since this would affect the way in which the MS was displayed in the web app: for members of partbook sets we wished to show the master or group shelfmark in the main MS listing, with the constituent parts of that set inset as a list below (perhaps as a collapsible list). If some members of the partbook set were lodged in a different library this would require a link to the other books.

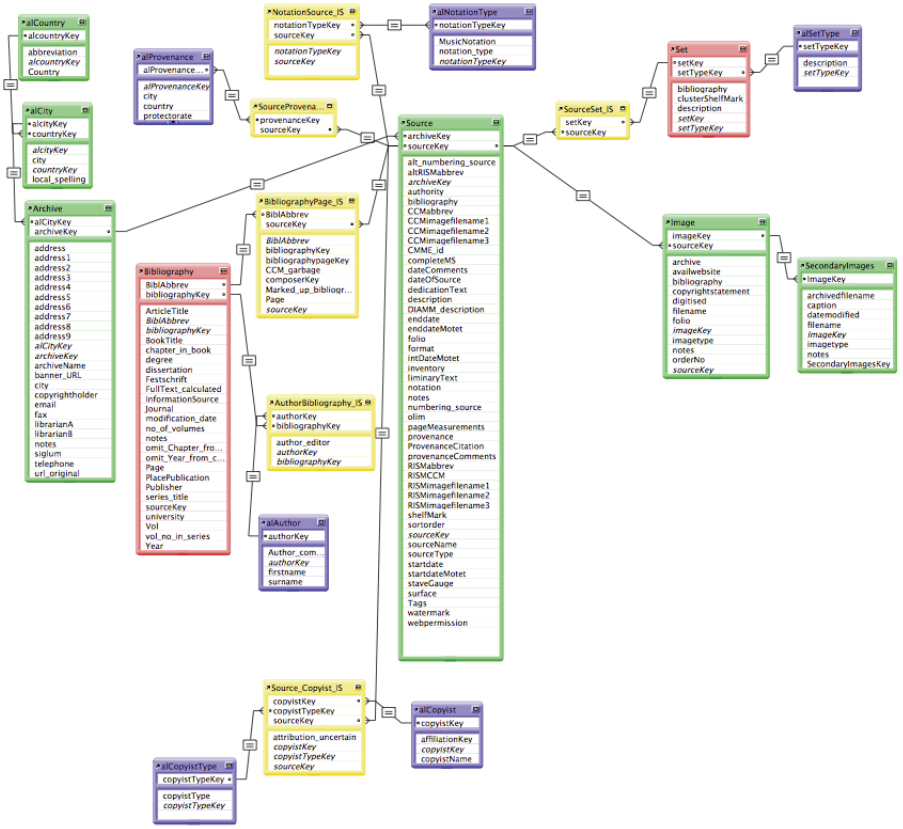
The same display was not however required for sources grouped together e.g. because they had been copied by the same scribe. These sources might never have been intended to remain together, and are usually widely dispersed. Their connection however is clearly important and needs to be manifest in some way visually. In this case each listing in the 'browse sources' display might show a link that would take the user to a list (or popup list) of the other manuscripts in that set.

One further facility was envisioned for this addition to the data, and that was the intention that at some point users might be able to come to DIAMM and search or study only a particular group of sources – perhaps a subset of the music manuscripts, but also in thinking about long-term sustainability of the project and the wider

exploitation of the web app we might want to include non-music collections, or sets of sources related to a particular research project within DIAMM. The set definition would allow us to do this. For this reason the *type* of set is absolutely crucial, and must always be defined when a manuscript set is created.

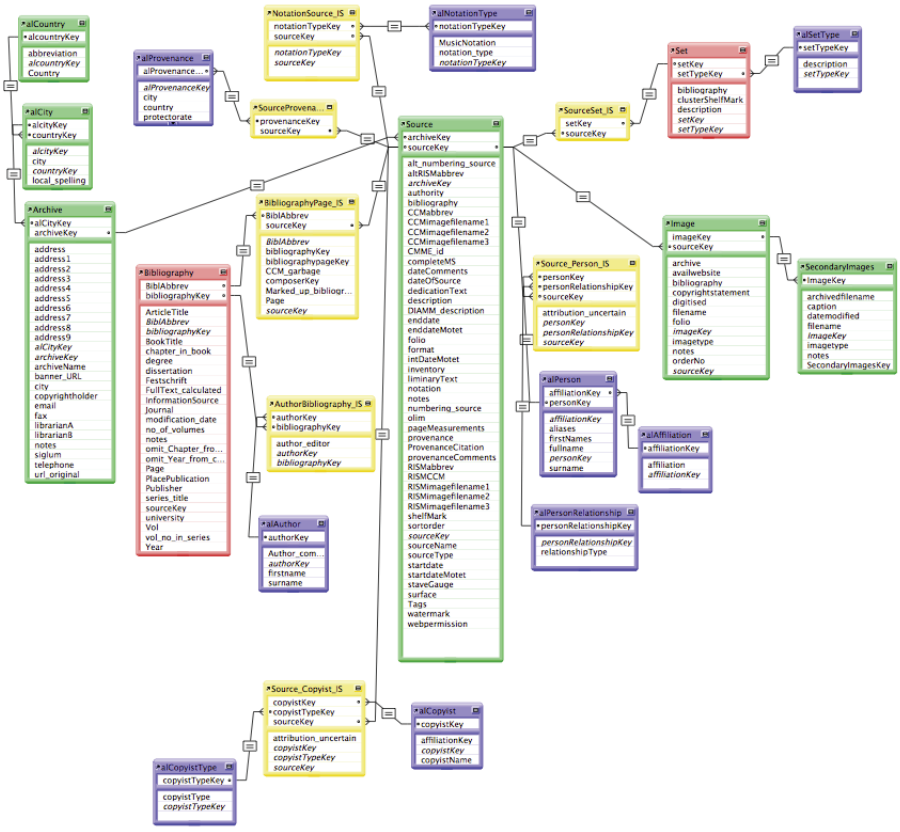


Once we started to address how manuscripts might be related to each other in non-musical as well as musical ways it was clear that in order to search efficiently on these criteria those relationships needed to be expressed by a series of authority lists, related to the **Source** table through intersection sets. The growing population of the database (which by now had far exceeded just those sources for which we had images) led to what had previously been managed in fields being migrated to authority lists – notably information about the type of **notation** used in the document. Although strictly this might not have needed to be treated this way, the authority list improves searching in the web app, but more importantly would allow the academic contributors to look at the source descriptions and bring terminilogy up to date across the board with this method of data management.



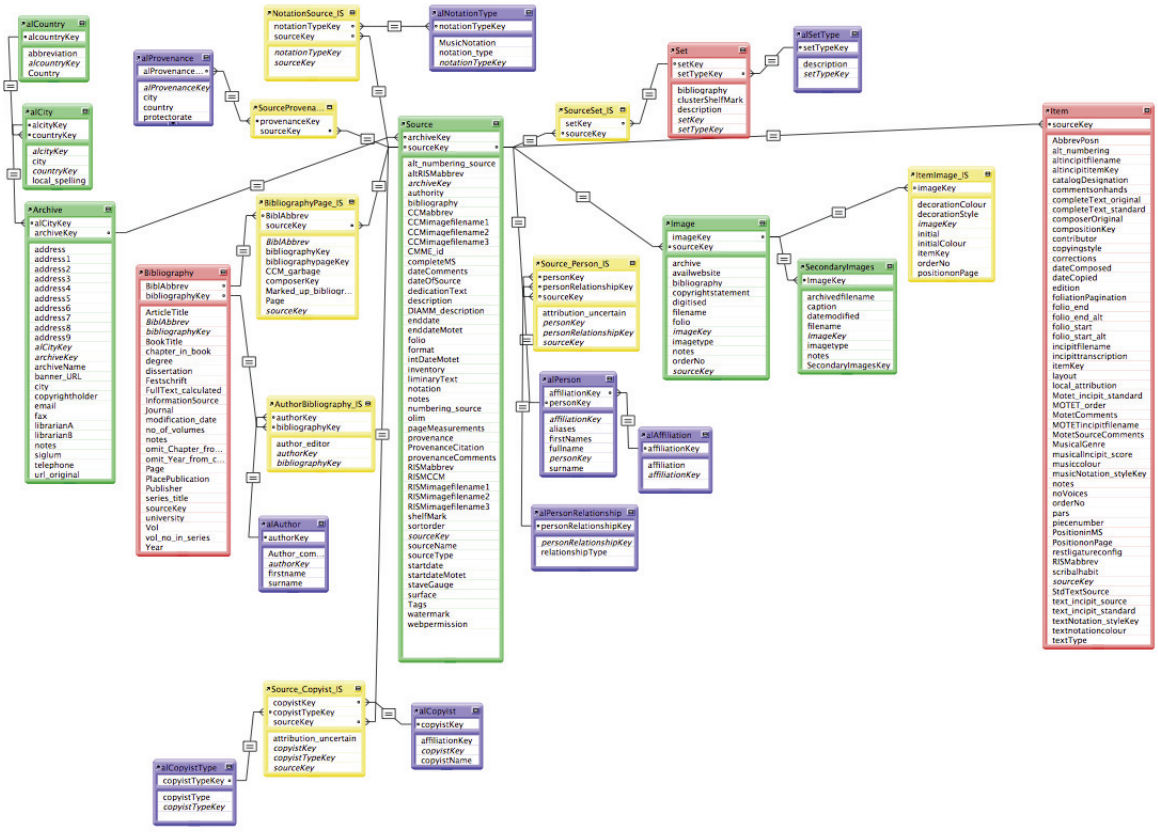
The discussion surrounding the categorisation of sources into sets gave rise also to the addition of an authority list to manage scribes (**copyists**), with a further authority list attached to the intersection set that would define the type of copying work (e.g. music, text, illumination etc.). The next stage added a '**Person**' table which was treated separately from the copyist list, since the relationship to the document had to be managed differently and for technical as well as taxonomic reasons needed to be kept separate.

The **Person** table deals not only with people but also with a wider definition of those related to a book: so one MS might have been owned by Abbot X, but another might have been owned by the Abbey of Y. It seemed unnecessary to differentiate here between a person or a place, and the name of the table was largely irrelevant. However, linking a person (or persons) to a MS required, as with the scribes, a definition of the type of relationship that the person had with the MS, giving rise to an authority list linked to the intersection set that linked **Source** to **Person**. Further, it became clear that another authority list was required that allowed us to list the **affiliation** of a person. So a MS might be owned by the monk Boniface, who was the abbot of a particular monstery (described in the **affiliation** table), who gave the MS to another monk on his death, also affiliated to the same monastery. The connection of both monks to the same record in the **Affiliation** table ensures the relationship between the two monks is made explicit, although the resulting data structure is slightly complex.



In terms of timing, these satellite tables actually post-dated the expansion of data content to increase the base granularity from the level of the **Source** (or manuscript) to that of the **Item** within the manuscript, but it is described here in relation to the base table since this makes it easier to understand the overall structure.

Since items within a manuscript (i.e. musical compositions) do not correlate with images (or pages within a source) a new table, **Item**, was introduced to the structure which enabled us to list the contents of a book. The **Image** table can at this point be considered to represent pages in a book (although it also includes pictures of bindings and other material related to a MS as a physical entity). An item may run over many pages, or a page contain many items, so an intersection set is used to connect the **Item** table to the **Image** table. The web app was still yielding information basically at the level of the source, since that was our primary unit at this point, particularly as the data sources were variable in the amount of information they supplied, so that the level of the source was the only one at which we could provide a parity of information, even though it would range in content detail between sources.



An examination was undertaken at this point of the type of data expressed in prose descriptions with a view to editing the descriptions we had so that all supplied the same level of information. The information might then be divided into separate fields so that it might appear under consistent headings: e.g. copyists, date, provenance, dimensions, contents, owners, binding, watermarks, gathering information etc. A brief study was undertaken of the way in which scholars had approached the description of these documents in the printed catalogues: one entry might discourse at length on the history and ownership of a manuscript and entirly omit any description of its biding, while another would give lengthy information about the binding and collation, but no information at all about ownership or copyists. Even descriptions of the same manuscript in the two different catalogues could show surprising disparities between even such information as might be expected to be consistent, such as measurements.

A single document description was analysed to show the different types of data that the prose supplied. The first and most obvious point to note was that because free prose is rich and complex in the way it expresses information, dividing that information into separate fields or headings might involve significant repetition, which in data terms is very inefficient. Returning to the basic description, this first analysis identified some 26 information areas. When that model was applied to other descriptions the number increased almost with every description that we examined, up to something in the region of 180 different descriptive areas. Clearly attempting to mark up the prose text, or divide it into fields was going to be extremely difficult, massively time-consuming, and possibly ultimately pointless, since the richness, continuity and overall effect of the prose description would be lost in the process.

More useful was the examination of the information provided by the RISM catalogue musical incipits. At this point musical incipits cannot be meaningfully searched (and early searchable transcription softwares require the reinputting of a mass of data), so providing the user with searchable information would involve the 'textualised' transcription of much of the data encapsulated in the music into text fields and tables. This formed the basis for what would later become the **Item** table and its satellite and related tables and authority lists. The analysis shown below concentrated only on the text parts of the information; the markup omits data such as clef, mensuration, key signature, layout or primary pitch manifest in the musical notation, many of which were subsequently transcribed into the database.

CAMBRAI, Bibliothèque Municipale, B. 1328 (F-CA 1328) (CaB)

21 folios from at least 4 parchment manuscripts dating from the 13th and 14th centuries, and now arbitrarily collected together under one signature. Ludwig thinks 16 folios come from the same codex of the second half of the 14th century. If this is so, the motet section on f. 17-20 and f. 11 and 12 belongs to the same source as the secular songs (Ballades, Rondeaux and Virelais) on f. 2, 4, 7-10 and 13-15. Quite clearly in a different hand and from different mss are f. 5 and 6 (which should proceed in the reverse of the present order), f. 3, and f. 16 and 21. Fol. 5 and 6 contain a three-part Agnus and the well-known Ballade *De ce que fol pensé*, which also appears as no. 23 in the main codex. The two folios in question employ a neater notation than the principal ms and their somewhat smaller page size meas. 225x165 mm. The Ars Nova notation is written on 8 red five-line staves per page (f. 5v in brown). The bottom of f. 6 is cut away. Fol. 16-21 are from a Gregorian codex and contain liturgical monodies on 9-10 red four-line staves per page. Fol. 3 meas. 225 X 158 mm. contains 4 known Rondeaux of Adam de la Hale in a pre-Franconian notation of the late 13th century with long-breve differentiation, c.o.p. ligatures and three-semibreve groups or breve + two-semibreve groups. As usual in this type of composition, there are 3 accolades of 3 red four-line staves per page in score with red or blue initials. Once again the present recto is really the verso and vice versa. Ludwig believes the mass section on f. 1 belongs to the main codex with the motets and secular songs. The motets are written in 2 cols., however, while the secular songs and mass pieces go straight across the page in the same Ars Nova notation. The various pages meas. as follows: f. 1: 305x237 mm., f. 2: 302x205 mm., f. 4: 244x185 mm., f. 7: 242x185 mm., f. 8: 303x208 mm., f. 9: 310x213 mm., f. 10: 316x220 mm., f. 11: 309x220 mm., f. 12: 303x227 mm., f. 13: 277x198 mm., f. 14: 275x187 mm., f. 15: 310x210 mm., f. 18: 283x233 mm. Fol. 17-20 form a consecutive pair of bifolios which are cut down at top and bottom, though f. 18 is complete at the top. The other pages are not quite as long. Fol. 19 and 20 are cut in half vertically, so that only the left col. of the motets remains. Since these 21 pages have been removed from various bindings, often without sufficient care, a number of pages which were formerly glued into bindings are now almost or completely illegible. These include f. 1v, 2v, 4v, 7, 8v, 9v, 11, 12, 13v, 14, 15. Finally, here are a few further details of the individual pages, which are now foliated in 19th century ink at t.r.r. Fol. 1 contains 11 (originally 12) red five-line staves per page and employs red initials. Fol. 2 employs red or blue initials and the same number of staves as f. 1. Fol. 4 is similar to f. 2 but cut down, so that only 8 staves remain. Fol. 7 is also a cut-down page like f. 4, doubtless from the same ms. Fol. 8 is a larger piece again, but upside down. The recto bears an old folio number viii at t.m.r. Initials are red or blue, and there are 11 red five-line staves per page. Fol. 9 employs some red notes, again on 11 red five-line staves. On f. 11 the top stave is cut away; there were originally 12 red five-line staves per page, as on f. 12. Fol. 13 has now only 10 red five-line staves per page with red or blue initials. Fol. 15 contains 11 red five-line staves, f. 15v 10. Fol. 17-20 contained originally 12 staves.

BIBLIOGRAPHY

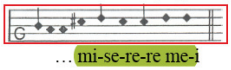
Music

AudaTu I-II, motet 26 (facsimile and transcription of *CA 1328* 38 after I-Tu); ApelF, no. 69 (transcription of *CA 1328* 44); BesselerS II, 254ff, 247ff, 250ff (transcription of *CA 1328* 37, 42 and 47); J. Chailley, *Adam de la Halle, Rondeaux à trois voix égales* (1942), nos. 1—4 (transcription of *CA 1328* 7-10); CoussH, pl. XXXI, 36, 35 and 34 (diplomatic facsimile of *CA 1328* 9-10, 18, 20 and 26); C. E. H. de Coussemaker, Notice *sur les collections musicales de la Bibliothèque de Cambrai*, in *Mémoires de la Société d'émul. de Cambrai* XVIII (1841), nos. 19, 20, 23, 24-25, 21, 22, 10-11, 12, 15, 16, 17, 18 and 26 (editions of texts of *CA 1328* 3, 4, 18, 19, 20, 21, 25, 26, 27, 28, 33, 34 and 36); GennrichR I, 54 ff (transcription of *CA 1328* 7-10); II, 50 (transcription of *CA 1328* 19); GüntherCM, no. 3 (transcription of *CA 1328* 37); GüntherZ, no. 6 (transcription of *CA 1328* 18); MachM I, Ballade 18 and Rondeau 7; III, motet 8 (transcription of *CA 1328* 35, 29 and 49); G. Reaney, *Early Fifteenth-Century Music* II (1959), 102 ff (transcription of *CA 1328* 24 after ms Oxford, Bodleian Library, Canonici misc. 213); SchradeM I, 82ff, 85 ff, 76 ff (transcription of *CA 1328* 39, 42 and 47); II-III, Rondeau 7, Ballade 18 and motet 8 (transcription of *CA 1328 29*, 35 and 49); Stäblein-Harder I, 66f, 132f, 139f (transcription of *CA 1328* 2, 14 and 41); WolfH I, 354ff (transcription of *CA 1328* 15 = 23); WolfS, pl. 81 (facsimile of *CA 1328* 19 after CH-BEb *421*)

Books and Articles

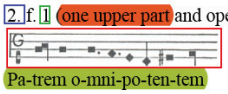
BesselerS I, 197ff; GüntherCM, XXIV ff; Jackson; LudwigA, 283ff; MachM II. 20f and passim; Stäblein-Harder II, 51, 78, 81, 138, 167f, 169

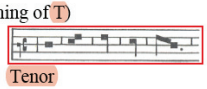
1. f. 1 (last line of a single voice with the following (*Expl.*)



… mi-se-re-re me-i

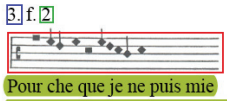
2. f. 1 (one upper part and opening of T)

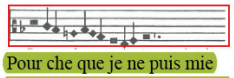
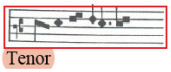


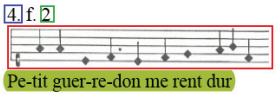
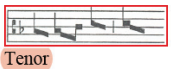
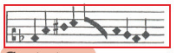
Pa-trem o-mni-po-ten-tem

Tenor

2a. f. 1v (Tr of no. 37, almost illegible)

3. f. 2



Pour che que je ne puis mie
ou vous m'a-vés, douche amie

Pour che que je ne puis mie
ou vous m'avés, douche amie

Tenor

4. f. 2

Pe-tit guer-re-don me rent dur

Tenor

Contratenor

5. f. 2v (mainly illegible; a three-part Virelai with T and Contratenor)

Hé, douls regars, dame de pris

6. f. 2v (illegible)

Tristour et merancolie

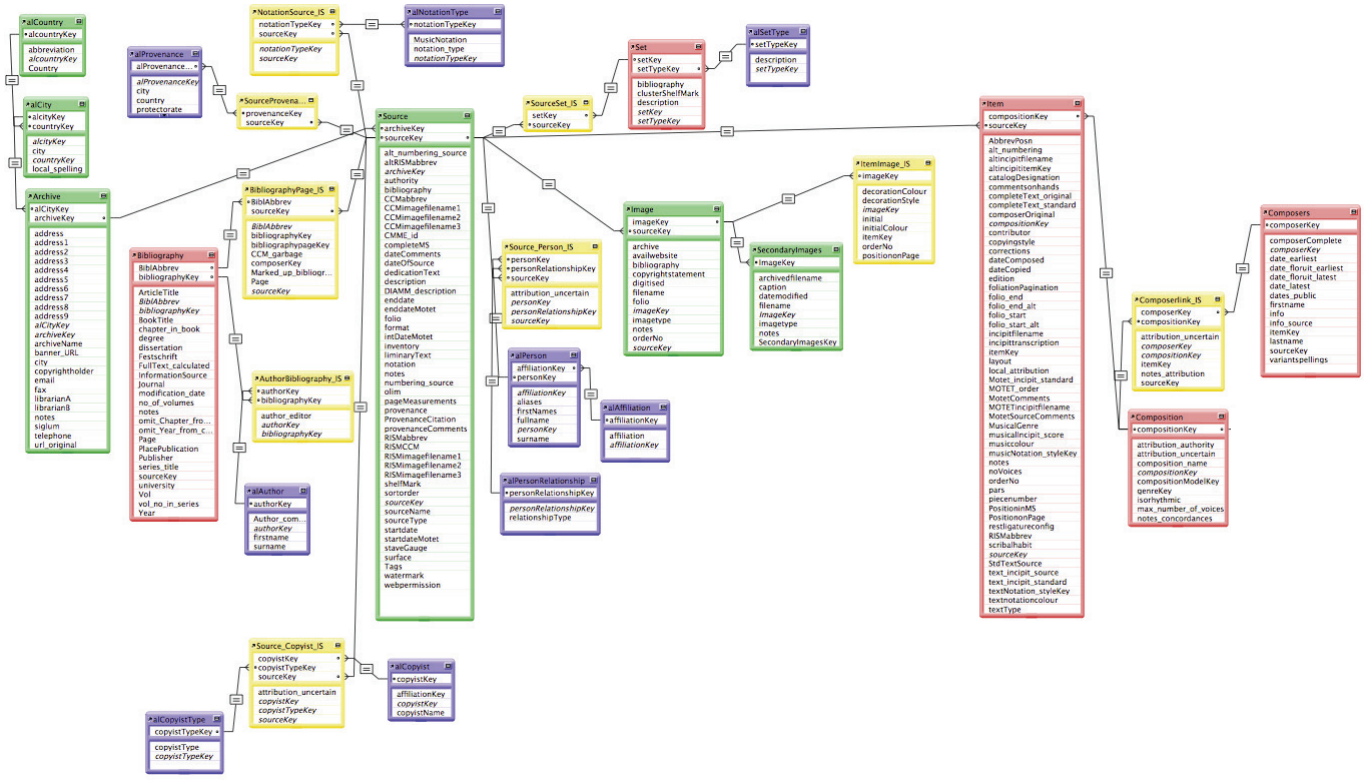
Key

city
library
shelfmark
RISM siglum for MS
internal RISM reference
date/period
surface/material
physical origins
physical make-up/gatherings/sections/divisions
condition/damage or disarrangement
genres
titles/text incipits
page measurement (always mm)
number of staves per page
number of lines per stave
number of accolades per page/no of staves per accolade
type of ruling (rastrum/hand) **
stave height **
initials on page (y/n) (historiated or plain) **
initial letters **
colour of initials **
notation type
scribal habit/characteristics/
rest/ligature configuration
foliation details
comments on individual hands

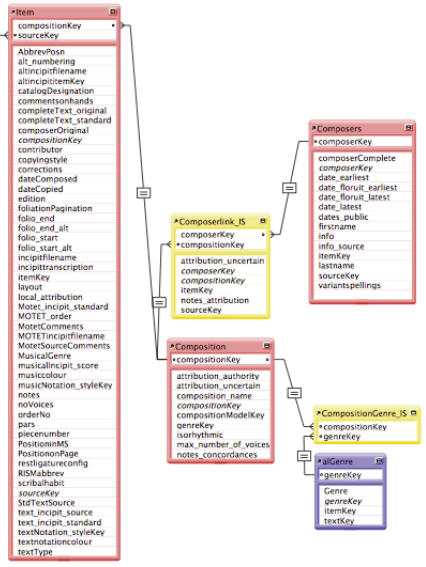
layout (score, choirbook, etc.)
musical incipit (jpeg/ note on legibility/direct to other MS for jpg)
composers
clefs*
mensurations*
key signature*
piece number
folio number
voice/part designations
number of voices/parts shown and/or in original work
editions of the music
facsimiles
bibliography
general notes

*extracted by eye from RISM incipit unless described in text
**information may have to be taken from image if not detailed in th text

Items represent an isolated instance of a composition – an intellectualised idea of a musical work based on all the known instances of it. In later repertories the composition achieved a fixed and final form when the composer chose to publish it, thus creating his own idealised version of it and discarding variant or sketch versions in favour of the final form. The idea of a final or finished form of a work was growing through the period represented by these manuscripts, but without mass-market publishing the concept of a single version of a work was undeveloped. Some works appear in one manuscript with four voices and in another with three (for example), with added parts possibly contributed by a different composer. This fluidity has led to considerable argument about the purpose of a 'modern edition' of a work in this period, which often draws on a number of sources to create a concatenated idealised version that did not represent any performance or transmission of the work that its composer might recognise. Regardless of the pitfalls of this modern view of musical composition, the existence of concordant sources for a work means that we need to find a way to express the fact that there are items which are versions or copies of other items. For this reason the **Item** table is 'governed' by a **Composition** table, which represents this idealised intellectual version of a work. Each iteration of the composition is connected to a master record in the **Composition** table.



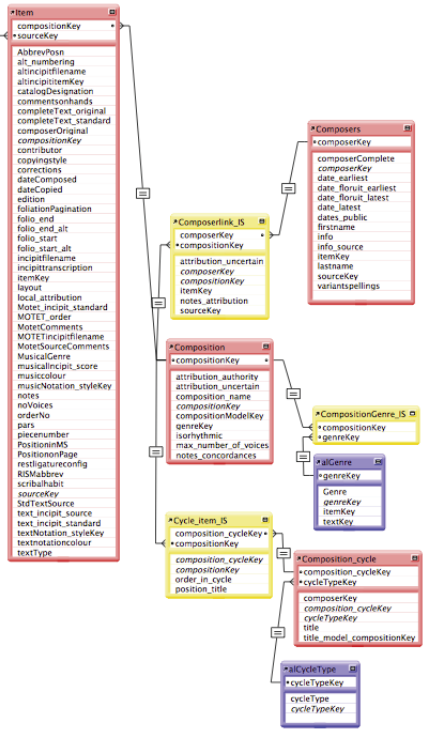
The **Composition** table is in turn linked via intersection set to a **Composer** table, which manages alternative or multiple attributions of a work to a number of composers. Each composition is also linked by intersection set to a **Genre** authority list, which categorises works in both general and specific terms, allowing any work to be allocated a number of genres.



The following diagrams, for the sake of clarity, deal mainly with a master table and its satellites generated following analysis of the otherwise unsearchable data provided by a musical incipit. At the time of writing this situation is changing, but since so much music would need to be typeset in order to create a searchable body of data, the DIAMM dataset relies on text searchability for the basic information, and access to images of transcribed incipits for a large part of the existing corpus, where the user can see information that cannot be usefully translated. The point about musical notation is of course that it is a system that expresses information which cannot be expressed in text, so any attempt to transcribe it in such a way is egregious.

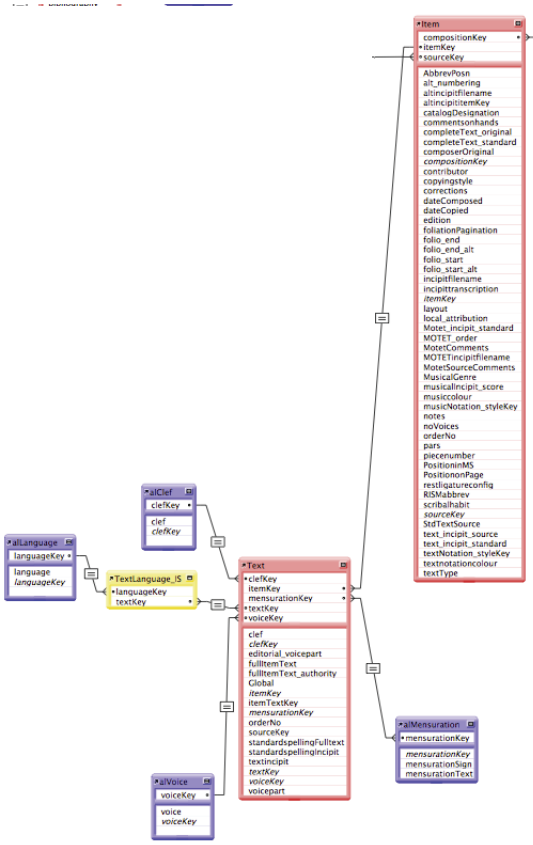
Like manuscripts, compositions can cluster together in sets or cycles. Sometimes these are sets of compositions gathered together because they are intended to be used together (such as music for a saint's day), or because a compiler feels they work well together for reasons of style or key. These compositions are naturally individual, but can be joined by being members of a set. Other compositions were designed to exist together: movements of a mass cycle for instance. Still other types of cycles exist where we found the same data management problem as we had with partbooks: originally listed as single items with constituent parts, we found that each partbook in a set had to be treated as individual sources.

The analog to that situation in the **Item** table was the 'single' musical work that was divided into sections. Although all those sections might be expected always to appear in the same order and complete whenever that composition was copied, this was not in fact the case. Sometimes a 'secunda pars' of a work would appear in another source as an independent work in its own right.



This gave rise to two problems: the simple practical one of how to express the fact that a composition might be part of a cycle within the database; but more difficult to manage, the problem of reading the data provided by a cataloguer: whereas cataloguer A might treat a piece and all its constituent pars as one work, cataloguer B might prefer to treat each pars as an individual work in its own right. Further, individual definitions of cataloguers might lead one to regard a section of a work as a pars while another would not. Often we did not know which method prevailed in our source data, so the content of this table (at the point of writing) may need significant editorial input. The simple quantity of data too makes this a thorny issue.

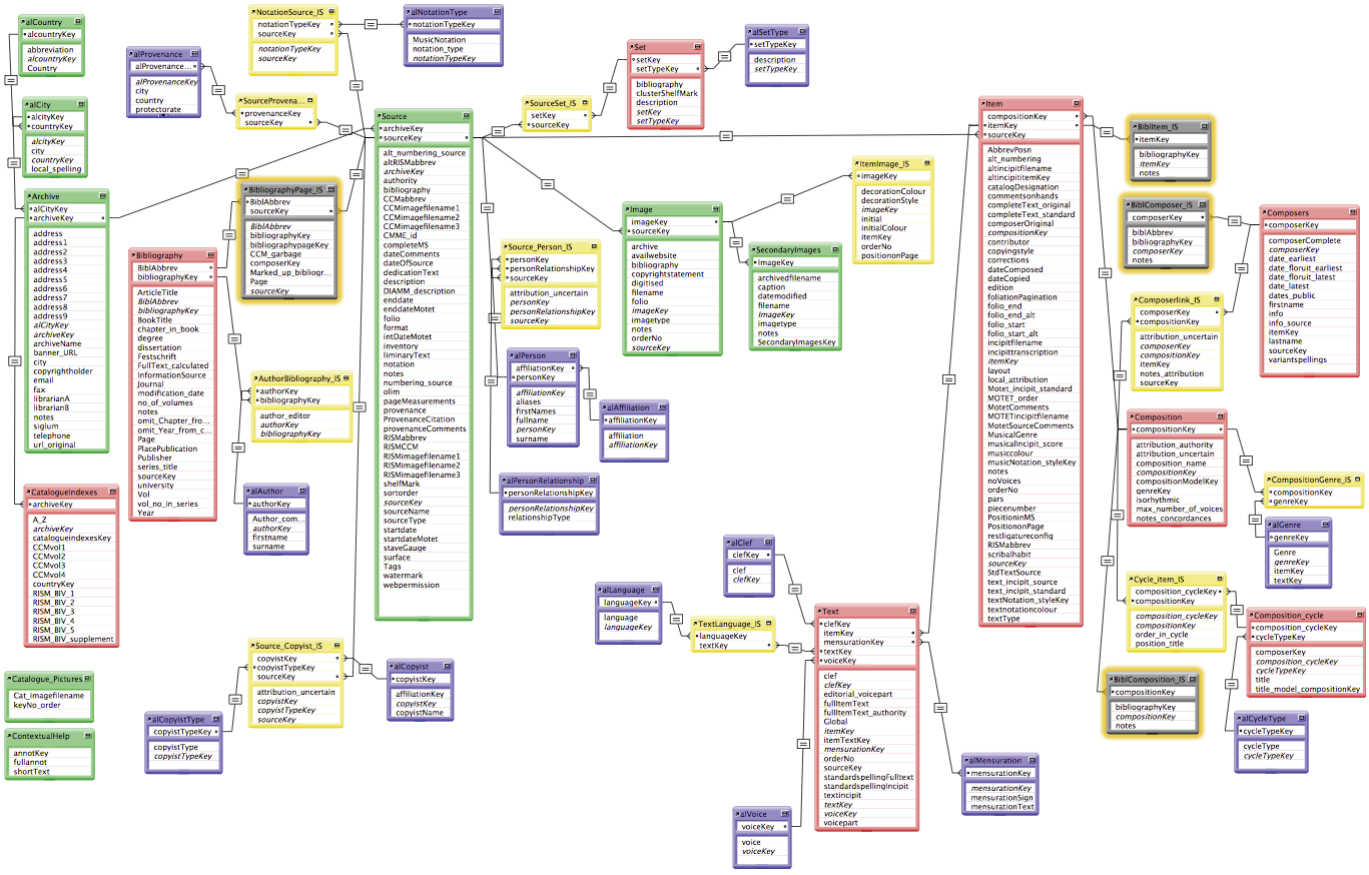
The **Composition** table thus accrued a further satellite: the **Composition_Cycle**. Connected to the **Composition** table by an intersection set, and also governed by an authority list that defined the type of cycle.



Returning to the **Item** master table and its expression in terms of a musical incipit, each item is clearly extremely rich in data, particularly material that might need to be searchable in the web app. The richest source of information is the text. Unlike later works, certain genres in this period might have different texts in each voice, and so the **Item** table was far from sufficient in itself to express the individual information about each voice part in the iteration of the item. Since an item might have a different number of voices in disparate sources this also made it necessary to differentiate the voice-part information from the overal **Item** information. A **Text** table was introduced to handle the content of each voice part, and that table required a number of authority lists to govern its content in turn.

The text underlay required the definition of language, and this is linked via intersection set as some texts (particularly troped texts) use more than one **Language**. Since the **Text** table represents a voice part (there should be a text record for each voice in an item) information about **clef**, **mensuration** and **voice** designation are given in this table. These are also managed with simple authority lists. No attempt is made to indicate changes of mensuration or clef during the course of a piece, only those shown at the start.

Finally, three intersection sets were added between the **Bibliography** table and other tables where individual bibliographical information about more granular content was available: **Composer**, **Item** and **Composition** (shown below in black and highlighted, along with the existing Source-to-Bibliography intersection). The existing intersection set betwen the **Source** table and **Bibliography** was not used for this purpose to retain clarity between the relationships. The intersection sets include notes fields that allow the user to indicate pages etc. This is the only relationship where the key numbers are not currently used to connect tables: the bibliography intersection sets were set up to allow users to link from a bibliographical abbreviation to the master bibliography record, and for this reason the relationship is currently managed via this field. Given the prevalence and widespread usage of the abbreviations it is not presently practical to change that connection to a simple numeric one, although the numeric connection is in place and is used by the SQL version of the database



A new strategy regarding data input and layout content will be adopted on conclusion of the 2011 data input work: the upload process is slow and clumsy, has to be done in a certain order, and is liable to problems if an upload fails that need to be fixed by a server administrator. Uploads of the entire dataset are therefore undesirable, particularly if they are being done simply to implement minor changes in one table, or even one record. When data input ceases therefore, a single master upload will take place, and then the content of the working layouts will be changed so that the fields users edit in the database will actually be fields in the SQL database, not FM fields. When corrections are done they will appear immediately in the web app on the live staging server and can be checked there, and then that version of the SQL data can be pushed to the live server.

This circumvents any upload from FM to SQL and eliminates the risk of loss or corruption of the master database following a crash. As this means that there would be no 'local' record of changes to the master dataset, the relevant fields in the FM database will become lookups that will copy data from the SQL fields, providing a local backup for the content.

The major exceptions to this will be the description and bibliography fields in the Source table. These fields are calculated and collate information from a number of other fields. Therefore edits to these fields will have to be copied manually to the SQL version. This should not be a problem, and can be scripted to a button-click.

2. CONTENT

2.1 Layouts

As described in 1.3 above the working database includes a number of relationships, table duplications and redundant fields that facilitate the creation of layouts that show information from a number of tables without the user having to keep a number of tables open at one time. Further non-essential fields (in data content terms) were introduced to allow the user to see lists within a form layout; these are global fields, which have the same value across the dataset. They are not essential to the functioning of the database but are essntial to the functioning of certain layouts. They have been removed from the **DIAMM_Shell** version of the database, as have most of the layouts used for data input in the working database.

The variety of different tasks undertaken with the content has led to a proliferation of layouts of varying complexity. An attempt was made to standardise down to three or four master layouts which showed all the information for a table and much of the information relating to it that was populated from the master table, but these layouts became too complex in many cases for clerical clerical staff to navigate and had to be slimmed down into custom layouts for specific tasks.

2.2 Data

The DIAMM data is centred around manuscript sources (known as **Sources**), which means that all other data, either more or less granular, radiates from that central point. Other databases of music from this period may take a composition (known in this dataset as an **Item**) or a composer as its starting point, and list manuscript sources where that composition (or where that composer's works) may be found. Still others may be library-based, and to some extent the DIAMM dataset is also library based, since the locations of the MSS are crucial to the core data about each source.

The starting point for the database is therefore the **Source** (manuscript) and where it is found.

The primary tables in this respect are:

Country
City
Archive
Source

Country and **City** are self explantory, and **City** is a subset of **Country**.

Archive is a subset of **Country**, and covers any type of repository from major national libraries to local archive offices and private collections of individuals. The document of interest may be the only document in the collection, or the only musical document, but is more often one of a reasonably large collection.

Source refers to the manuscript within the library collection. It may range in extent from a complete bound book with many pages, to a group of fragmentary leaves perhaps bound in a modern guard book with other unrelated documents, and sometimes to fragments of parchment or paper no larger than a postage stamp. In each case this is considered a source. Each source usually has a shelf-mark designation which identifies it to the librarian, though often in private collections there is no library mark, and perhaps only a colloquial name by which the document is known. Whatever the designation this is the form under which a visitor would ask to see the source, and must be kept accurate and up-to-date as far as possible.

A significant proportion of libraries and collections have undertaken recataloguing during which shelf marks were changed. Surprisingly, few archives that have recatalogued their collections kept a cross-reference document listing old and new shelfmarks, and DIAMM may in many cases be the only resource that does this. Many of the printed catalogues dealing with our sources predate changes in shelfmarking systems, so it is important for the database to keep a record of previous shelf marks to ensure users coming to the resource from one

of the older printed catalogues can still find the manuscript they seek. This is managed using the olim field in the Source table

More complex are the group of sources that have moved from one archive to another: sometimes physically the sources are actually in the same place, but the archive has been renamed, but in some cases, particularly with German and Polish libraries, where manuscript collections moved around Europe during the Second World War, the manuscripts have changed location. Sometimes the original shelf-mark is retained, but often a new shelfmark is allocated.

In the case of books that have actually or apparently moved/changed ownership, a record is created for both sources for the books, the out-of-date location pointing to the new (current) location.

2.3 Tables and Fields

There are three types of table:

Master or Primary tables — these tables are the master reference tables and contain multiple fields in each record. Because of the way that the database is structured some master tables behave like authority lists or intersection sets. Their formal designation is not important, but classifying them in this way allows the user to better understand the structure of the information provided.

Authority lists — simple lists of data that would otherwise be repeated in a master table. They provide the user with a predefined set of data from which to choose, e.g. a list of clefs, a list of countries, a list of authors. These may be linked directly to Master tables, or connected via an intersection set.

Intersection sets — These are used in cases where the relationship between a master table and an authority list is many-to-many rather than one-to-many. The relationship betwen Text and Language might be considered as one-to-many (i.e. one language for each text record), but in fact some texts are constructed using more than one language, so these two tables are linked by an intersection set. Intersection sets also allow us to create information about the connection between information in two tables. For example, the intersection set between master tables and the bibliography connect a source, item, composition, composer etc to the bibliographical item, but the intersection set indicates the page relevant to the master item.

The following information is colour coded:
Table names
Field names
Link field names (used in relationships)
Fields in the Filemaker database that are not included in the SQL database

The following lists show the **table** and field content of the working DIAMM database (filename: **DIAMM**) with a short explanation of the content of the table; Field names in this colour are those omitted from the **DIAMM_Shell** database (which eliminates redundant fields and tables). Calculation fields in **DIAMM** are Text only in **DIAMM_Shell**.

The database can be copied and renamed DIAMM_RUBBISH for the purpose of creating a practice database: the name of the file is crucial as the use of the word 'RUBBISH' in the title triggers a conditional formatting script that will automatically place a warning flag on all layouts reminding users that they are not working in the main database.

Table names prefaced by 'al' are authority lists; Table names suffixed by '_IS' are intersection sets; field names suffixed '_g' are global fields. Key number fields (usually these are the fields are the ones used for creating relationships) are suffixed 'Key'.

alAffiliation

This table lists the affiliations of persons linked to a manuscript (i.e. listed in the manuscript description). At present it connects to the **Source** *table, but can be used to refer to to individual works. It is linked to the source database via the person authority list,* **alPerson**.

Field Name	Field Type	Formula / Entry Option
affiliation	Text	
affiliationKey	Number	Serial Number on creation with Current Value: "424" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
g_Key	Number	Indexed, Auto-enter: "99", Allow user to override validation, Required value, Only allowvalues of type: "Numeric Only"

alAuthor

This table gives a list of names of authors of secondary source material. It is related to the **Bibliography** *database through an intersection set which gives the relationship of the person named to the bibliographical item (i.e. author; editor; compiler; publisher; Festschrift dedicatee; translator; collaborator; later editor).*

Field Name	Field Type	Formula / Entry Option
Author_complete	Calculation	(Text) = *surname & If (IsEmpty (firstname) ; "" ; ", " & firstname)*
authorKey	Number	Serial Number on creation with Current Value: "2085" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
firstname	Text	
g_Key	Number	Indexed, Auto-enter: "99", Allow user to override validation, Required value, Only allowvalues of type: "Numeric Only"
surname	Text	

alCity

The listing for archives and libraries has two parent tables, **alCountry** *and* **alCity**. **alCity** *acts as an intersection set between* **Archive** *and* **alCountry**. *It simply lists all cities for which the database lists a library. 'City' can mean a much smaller location such as a village if the library is not located in a major urban centre.*

Field Name	Field Type	Formula / Entry Option
alcityKey	Number	auto-enter serial, required value, unique
city	Text	name of city
countryKey	Number	Key field used in relationship to alCountry
g_Key	Number	Indexed, Auto-enter: "99", Allow user to override validation, Required value, Only allowvalues of type: "Numeric Only"

alClef

A simple authority list with lists all the clef types in use and relates to the **Text** *table, since that table effectively represents the individual voice parts of a piece, even when these are untexted.*

Field Name	Field Type	Formula / Entry Option
clef	Text	
clefKey	Number	Serial Number on creation with Current Value: "87" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
g_Key	Number	Indexed, Auto-enter: "99", Allow user to override validation, Required value, Only allowvalues of type: "Numeric Only"

alCopyist

The copyist authority list links a list of copyists or scribes to the **Source** *table via an intersection set (* **Source_Copyist_IS**) *since one copyist may have worked in a number of different sources and vice versa. The copyist list has been treated separately from the list of persons associated with manuscripts because copyists are defined differently, since they may have undertaken differ-*

ent types of copying activity (see **alCopyistType**). *As well as being defined as a particular type of copyist they may also require an affiliation, so the relationship is more complex than that of 'person' which only requires a name and then a relationship type.*

Field Name	Field Type	Formula / Entry Option
affiliationKey	Number	
copyistKey	Number	Serial Number on creation with Current Value: "375" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
copyistName	Text	
g_Key	Number	Indexed, Auto-enter: "99", Allow user to override validation, Required value, Only allowvalues of type: "Numeric Only"

alCopyistType

This authority list allows the copyist of a source to be classified as a copyist of music, text, indexer, liminary text, illuminator or text and music together. It relates to the copyist list via the **Source_Copyist_IS** *since the same copyist may undertake different types of activity in a different manuscript. This authority list may appear redundant, since the information (regardless of its repetitive nature) could simply have been entered in the* **Source_Copyist_IS**, *but creating this as an authority list table was done to facilitate searching in the webapp.*

Field Name	Field Type	Formula / Entry Option
copyistType	Text	
copyistTypeKey	Number	Serial Number on creation with Current Value: "7" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"

alCountry

The country list refers only to the list of countries in which manuscripts are presently located. This list could not be used also as an authority list for the provenance of manuscripts, since that list would include countries that no longer exist, or regions that are now part of another country. This created a problem in presenting the data in the webapp, since when a user was browing for manuscripts by country/city they would see an entry for e.g. Burgundy, which was nonsense.

Field Name	Field Type	Formula / Entry Option
abbreviation	Text	abbreviated form of the country name
alcountryKey	Number	auto-enter serial, unique
Country	Text	full name of country

alCycleType

When compositions are linked together it is necessary to define what type of connection links them: e.g. a group of motets may actually be individual parts of one larger motet, or may be individual motets linked together as a group by the composer or a later editor/publisher. Individual mass movements form part of a different type of cycle. The table defines the type of cycle to which a composition belongs (if it is part of a cycle). This is a short list and could have simply been managed as a limited set of values in the **Cycle_item_IS** *table, but has been managed here to facilitate web searching.*

Field Name	Field Type	Formula / Entry Option
cycleType	Text	
cycleTypeKey	Number	Serial Number on creation with Current Value: "5" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
g_Key	Number	Indexed, Auto-enter: "99", Allow user to override validation, Required value, Only allowvalues of type: "Numeric Only"

alGenre

The definition of the content of this table is a little vague — a genre can be defined either as something very broad such as 'liturgical work' or something more specific such as 'Motet' or 'Mass movement'. The table is used here to provide both broad and specific categories that are applied to individual compositions and under which users might wish to search. Because a composition may therefore belong to more than one of these categories it is related to the composition table via an intersection set. A

work entitled Kyrie might therefore belong to the following 'genres': Kyrie, Mass proper, Liturgical work, Canon. It has also been used to group works in which a common text is used that has been set many times, such as Alma Redemptoris Mater, which is not a genre in itself, but the grouping is useful for searching purposes. At present more than 90 'genres' are defined in this table. It might simplify matters to redefine this table as the most granular type of work, with authority lists 'above' it in the hierarchy that would allow lower-level genre types to be grouped under that heading; e.g. ***Kyrie*** *is a child of* ***Mass proper****, which is in turn a child of* ***Liturgical Work****. However that would present a problem with* ***Canon*** *- where would that level of definition appear?*

Field Name	Field Type	Formula / Entry Option
g_Key	Number	Indexed, Auto-enter: "99", Allow user to override validation, Required value, Only allowvalues of type: "Numeric Only"
Genre	Text	
genreKey	Number	Serial Number on creation with Current Value: "101" Increment: "1", Required value, Unique values only, Only allow values of type: "Numeric Only"
itemKey	Text	Lookup: When an entry in "alGenre::genreKey" is made, copy the first matching value from "Item::itemKey" If no match: "Do not copy" don't copy "Item::itemKey" if empty

alLanguage

A relatively simple authority list, giving the language of the text underlaid in a single voice part of a composition. This had to be linked to the **Text** *table via an intersection set as some single voice parts have texts that are written in more than one language (distinct from a work such as a Motet, where each voice part may be in a different language—that usage is dealt with by each voice part (or Text) having its own language designation).*

Field Name	Field Type	Formula / Entry Option
language	Text	
languageKey	Number	Serial Number on creation with Current Value: "12" Increment: "1", Allow user to override validation, Unique values only, Only allow values of type: "Numeric Only"

alMensuration

This table provides the first mensuration for a voice part that appears in a work. It is linked to individual voice parts of individual instances of a work, since different voices may use a different mensuration, and a work may be copied using a different mensuration in a second source. We did not attempt to define mensuration changes that occur during the course of a piece. This would make the content extremely complex, since it might be considered necessary to define at what point a mensuration changes. That information was considered unnecessary for this level of data presentation.

Field Name	Field Type	Formula / Entry Option
mensurationKey	Number	Serial Number on creation with Current Value: "29" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
mensurationSign	Text	
mensurationText	Text	

alNotationType

A simple list of types of ***musical*** *notation in use in a manuscript. It is linked both to the* **Source** *table and the* **Item** *table (via intersection sets), which allows more than one notational type to be defined for each MS, and for individual items to have their notation specified (although that table is not yet populated). As with the genre list above this list contains items of varying granularity, based on the information available. In some cases a term as general as 'mensural' is the only information, while in others information such as 'black full and black void mensural' and 'minim replaces undifferentiated semibreves' is available. The content of this list is based solely on the descriptions provided by the printed catalogues, and some 80 musical notation types are presently defined.*

Field Name	Field Type	Formula / Entry Option
alNotationTypeKey	Number	Serial Number on creation with Current Value: "87" Increment: "1", Allow user to override validation, Required value, Unique values only
g_Key	Number	Indexed, Auto-enter: "99", Allow user to override validation, Required value, Only allowvalues of type: "Numeric Only"

notation_type	Text

alPerson

The **Person** *table lists names of people associated with manuscripts who were not copyists. These could range from an owner or dedicatee to the person who discovered the book in modern times. It does not however include the names of composers found in the book, unless a composer was also an owner. It is linked to the* **Source** *table by an intersection set (***Source_Person_IS***) which is informed by a further authority list,* **alPersonRelationship***, which defines the relationship that the person had with the manuscript.* **alPerson** *also has an authority list,* **alAffiliation***, which lists any connections that the person had e.g. with an institution. So Johannes Barbla may be listed as the owner of a book, and his affiliation would be 'Chaplain of St Catherine's Choir in Aachen'. The table's content definition is broader than simply individual people (despite its name), to allow more general connections to be expressed, so that a 'person' may be the Church of St Gudule, for whom the MS was commissioned, and who owned the book.*

Field Name	Field Type	Formula / Entry Option
affiliationKey	Text	
fullname	Text	
g_Key	Number	Indexed, Auto-enter: "99", Allow user to override validation, Required value, Only allowvalues of type: "Numeric Only"
personKey	Number	Serial Number on creation with Current Value: "1998" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"

alPersonRelationship

As describe in **alPerson** *above, this table defines the relationship that a person or institution (as broadly defined in* **alPerson***) had with a manuscript or source. There are currently 30 relationshps defined including owner, dedicatee, dedicator, binder, patron, author (of treatise in which music is interpolated), used as model for decoration style, discoverer, keeper, translator etc.*

Field Name	Field Type	Formula / Entry Option
g_Key	Number	Indexed, Auto-enter Data = "99", REquired Value, Numeric Only, Allow override
personRelationshipKey	Number	Serial Number on creation with Current Value: "34" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
relationshipType	Text	

alProvenance

The provenance of a book is one of the most important pieces of information about its history and content. This authority list is based on the level of information that is usually available about any manuscript: it's country of origin. The majority of manuscripts have no more detailed knowledge than that. The country of origin can include countries that no longer exist with modern boundaries, or which do not fit country boundaries as we know them now, such as Burgundy or the Low Countries. In order to allow a more finely-grained defintion of the origins of a book where that is known this authority list links to the **Source** *table via an intersection set (***SourceProvenance_IS***) which allows the inclusion of data about the region within a country, city, and/or institution within a city when that is also known.*

Field Name	Field Type	Formula / Entry Option
alProvenanceKey	Number	
country	Text	

alSetType

Sets allow manuscripts to be grouped, but the type of grouping needs to be defined as there are a number of different constructs that scholars use to connect mansucripts. One type is that of the intentional origin of a book, such as members of a set of partbooks. Another type links fragments that are now geographically separated, but belonged originally to the same MS, while another type is a modern connection made between groups of MSS for geographical or other reasons. The list currently is fairly short: partbooks; fragments that form part of one MS or collection; Special Music collection (e.g. Machaut MSS, Alamire scriptorium MSS); copyist or scribal concordance; single source bound in separate volumes. A further definition, Special non-Music collection, allows the database to include groups of MSS not related to the remit of the DIAMM project, but display these MSS using the DIAMM webapp.

Field Name	Field Type	Formula / Entry Option
description	Text	

setTypeKey	Number	Serial Number on creation with Current Value: "8" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"

Altincipit

This table creates a connection between the **Item** *table, in which a field links the item to a musical incipit file, and secondary incipits cited in the original incipit information. To save space and typesetting costs RISM would often cross refer from the instance of a work in one manuscript to its listing for that work in the entry for another MS—hence: 'incipit as for xxxx'. This table gives the alternative source for the incipit information and is a self intersection set for the* **Item** *table: it links to another entry in the same table. It looks up the*

Field Name	Field Type	Formula / Entry Option
AbbrevPosn	Text	
altincipitfilename	Calculation	(Text) Unstored, from Altincipit, = *Item::incipitfilename*
altincipitKey	Number	Serial Number on creation with Current Value: "2617" Increment: "1", Required value, Unique values only, Only allow values of type: "Numeric Only"
altitemKey	Number	Lookup: When an entry in "AbbrevPosn" is made, copy the first matching value from "itemKey" If no match: "Do not copy" don't copy "itemKey" if empty
itemKey	Number	Lookup: When an entry in "Altincipit::AbbrevPosn" is made, copy the first matching value from "Item::itemKey" If no match: "Do not copy" don't copy "Item::itemKey" if empty

alVoice

This table defines the name of a voice or individual part of a polyphonic composition. Often parts have no names and therefore are not defined, but this allows both the definition of voice names (e.g. Cantus, Tenor, Motetus) and unnamed parts where required (e.g. I, II, III…), and uses the terminology as given in the original source, rather than a standardised form. There are currently over 70 defined voice parts.

Field Name	Field Type	Formula / Entry Option
g_Key	Number	Indexed, Auto-enter: "99", Allow user to override validation, Required value, Only allowvalues of type: "Numeric Only"
voice	Text	
voiceKey	Number	Serial Number on creation with Current Value: "94" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"

Archive

This is the main table listing details of modern archives in which the MSS in the **Source** *table are housed. It gives practical information such as the address and contact details for the library, but it also gives internal information that is not displayed online such as the status of negotiations for licensing and/or information about the progress of digitization activity. It also gives the correct copyright statement to be shown with MSS from that collection. The address fields involve various types of compilation formats (the fields are not included in* **DIAMM_Shell***) to allow the data to be inserted into form letters generated by the database. There are a number of other fields, particularly calculation fields which are used to facilitate letter-writing and other internal activities.*

Field Name	Field Type	Formula / Entry Option
_PracticeDBwarning	Calculation	(Text) Unstored, from Archive, = TextColor (If (Right (Get (FileName) ; 7) = "RUBBISH" ;"YOU ARE IN THE PRACTICE DATABASE" ;""); RGB (0 ; 0 ; 0))
address	Calculation	(Text) from Archive, = *librarianA* & ¶ & *address1* & ¶ & *address2* & ¶ & *If (IsEmpty (address3) ; "" ; address3* & ¶) & *If (IsEmpty (address4) ; "" ; address4* & ¶) & *If (IsEmpty (address5) ; "" ; address5* & ¶) & *If (IsEmpty (address6) ; "" ; address6* & ¶) & *If (IsEmpty (address7) ; "" ; address7)*

address1	Text	
address2	Text	
address3	Text	
address4	Text	
address5	Text	
address6	Text	
address7	Text	
address8	Text	
addressrunon	Calculation	(Text) linear version of address used in inputting to google search engine in automated search for library URL or map; from Archive, = *address1* & " " & *address2* & " " & *address3* & " " & *address4* & " " & *address5* & " " & *address6* & " " & *address7* & " " & *address8*
alCityKey	Number	
alCountryKey	Number	Lookup: When an entry in "Archive::alCityKey" is made, copy the first matching value from "alCountry::alcountryKey" If no match: "Do not copy" don't copy "alCountry::alcountryKey" if empty
appendixofsources	Calculation	(Text) Unstored, from Archive, = *Source::shelfMark alone* & " *(*" & *Source::folio* & *");* "
archiveKey	Number	Serial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
archiveName	Text	
availableOnWebsite	Text	
banner_URL	Text	
CDcopiessent	Text	
CDschecked	Text	
city	Text	
copyrightholder	Text	
correspondencestatus	Text	
currentstatus	Text	
daysrequired	Number	
DONE	Text	
email	Text	Indexed
emailed	Text	
fax	Text	Indexed
ftp	Text	
fulladdressrunon	Text	
furthercorrespondence	Text	
g_Key	Number	Auto-enter calculation: from Archive, = "99", replaces existing value, Allow user tooverride validation, Required value, Only allow values of type: "Numeric Only"
g_libraryName	Text	Global
imagesarchived	Text	
imagesfromthisarchive	Number	Indexed
imagesordered	Text	
imagesreceived	Text	Indexed
invoicepaid	Text	
invoicequeried	Text	
invoicereceived	Text	
letter_content	Calculation	(Text) <The full text of this field is not displayed here, as it changes depending on the needs of the letter to be written to the library. This is a calculated field that draws on the list of sources for the library and is

		dependent on information provided by other fields.>
Phase	Text	
postprocessingdone	Text	
preliminary	Text	
readyforvisit	Text	
replyreceived	Text	
requiresPATC	Text	
responsefrom	Text	
responsefromdate	Text	
salutation	Text	Indexed
SelfGlobal	Number	Global
siglum	Text	Indexed
someoutstanding	Text	
sourcesfromthisarchive	Number	[of MSS in the archive in which we are interested]
statuslist	Calculation	(Text) from Archive, = city & " " & archiveName & ": " & currentstatus
telephone	Text	
telephoned	Date	
totalcity	Text	
totaldays	Number	[number of days required to digitize the sources on our list]
totalimages	Number	
updated	Date	Auto-enter: "Modification Date"
url	Calculation	(Text) Indexed, from Archive, = "http://" & url_original
url_original	Text	
visitdates	Text	
worthprodding	Text	
written	Date	

AuthorBibliography_IS

The intersection set between the **alAuthor** *and* **Bibliography** *tables that defines the relationship of an author with the book in the bibliography (i.e. author; editor; compiler; publisher; Festschrift dedicatee; translator; collaborator; later editor).*

Field Name	Field Type	Formula / Entry Option
authorBibliography_IS_KEY		NumberSerial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
author_editor	Text	
authorKey	Number	Lookup: When an entry in "AuthorBibliography_IS::authorKey" is made, copy the first matching value from "alAuthor::authorKey" If no match: "Do not copy" don't copy "alAuthor::authorKey" if empty
bibliographyKey	Number	Lookup: When an entry in "AuthorBibliography_IS::bibliographyKey" is made, copy the first matching value from "Bibliography::bibliographyKey" If no match: "Do not copy" don't copy "Bibliography::bibliographyKey" if empty

BiblComposer_IS

Links the **Bibliography** *table to the list of composers.*

Field Name	Field Type	Formula / Entry Option
biblAbbrev	Text	
biblComposer_IS_KEY	Number	Serial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
bibliographyKey	Number	Lookup: When an entry in "BiblComposer_IS::biblAbbrev" is made, copy the first matching value from "BiblComposer_link::bibliogra-

		phyKey" If no match: "Do not copy" don't copy "BiblComposer_link::bibliographyKey" if empty
composerKey	Number	Lookup: When an entry in "BiblComposer_IS::composerKey" is made, copy the first matching value from "Composers::composerKey" If no match: "Do not copy" don't copy "Composers::composerKey" if empty
notes	Text	

BiblComposition_IS

Links the bibliography to the list of compositions. The 'blue dog' field allows users to flag up potential problems with the data in a particular record.

Field Name	Field Type	Formula / Entry Option
biblAbbrev	Text	
biblComposition_IS_KEY	Number	Serial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
bibliographyKey	Number	Lookup: When an entry in "BiblComposition_IS::biblAbbrev" is made, copy the first matching value from "Bibl_Edition::bibliographyKey" If no match: "Do not copy" don't copy "Bibl_Edition::bibliographyKey" if empty
blue dogs	Text	
compositionKey	Number	Lookup: When an entry in "BiblComposition_IS::compositionKey" is made, copy the first matching value from "Composition::compositionKey" If no match: "Do not copy" don't copy "Composition::compositionKey" if empty
notes	Text	

Bibliography

Contains the modern bibliographical resource linked mainly (at this stage) to manucript descriptions. The field names are reasonably self-explanatory. The table has only one satellite table, **alAuthor**, *linked to it via an intersection set. The* **Bibliography** *table is however linked to a number of other tables, all of which may need to refer to a bibliographical item. The following tables connect to the* **Bibliography** *database via intersection sets:* **Source**, **Item**, **Composers**, **Composition**, **Text**, **Model_text**. *Although not currently linked to other tables, it should probably in the future also be linked to* **alCopyist**, **alPerson** *and* **alNotationType**. *The author fields defined here repeat the information available in the alAuthor table to facilitate the calculation that generates the full bibliographical citation. The BiblAbbrev field is in this case used as the linking field to the other tables, since this is the form of the book that is usually cited in catalogue entries.*

Field Name	Field Type	Formula / Entry Option
_PracticeDBwarning	Calculation	(Text) Unstored, from Archive, = TextColor (If (Right (Get (FileName) ; 7) = "RUBBISH" ;"YOU ARE IN THE PRACTICE DATABASE" ;""); RGB (0 ; 0 ; 0))
ArticleTitle	Text	
Author1Firstname	Text	
Author1Surname	Text	
Author2Firstname	Text	
Author2Surname	Text	
Author3Firstname	Text	
Author3Surname	Text	
Author4Firstname	Text	
Author4Surname	Text	
BiblAbbrev	Text	Allow user to override validation, Unique values only
bibliographyKey	Number	Serial Number on creation with Current Value: "4459" Increment: "1", Required value, Unique values only, Only allow values of type: "Numeric Only"
BookTitle	Text	
chapter_in_book	Text	

degree	Text	
dissertation	Text	
DONE	Text	
editing_history	Calculation	(Text) from Bibliography, = *If (IsEmpty (logged_in_as) ; editing_history ; Get (AccountName) & " " & modification date & ¶ & editing_history)*
Editor1Firstname	Text	
Editor1Surname	Text	
Editor2Firstname	Text	
Editor2Surname	Text	
Editor3Firstname	Text	
Editor3Surname	Text	
Editor4Firstname	Text	
Editor4Surname	Text	
Festschrift	Text	
FINDfield	Text	Global
FullBibliographyText	Text	
FullText_calculated	Calculation	(Text) Unstored, from Bibliography, = *If (IsEmpty (DONE); FullBibliographyText ;TextFont (TextSize (If (Length (Author1Surname) ≥ 1 and Length(Editor1Surname) ≥ 1 ;If (IsEmpty (Author1Surname) ; "" ; Author1Surname & ", " & Author1Firstname) &If (IsEmpty (Author2Surname) ; "" ; "; " & Author2Surname & ", " &Author2Firstname) &If (IsEmpty (Author3Surname) ; "" ; "; " & Author3Surname & ", " &Author3Firstname) &//translator info//If (IsEmpty (Translator1Surname) ; "" ;If (IsEmpty (Author1Surname) ; "(trans. " & Translator1Firstname & " " &Translator1Surname & ")" ;" (trans. " & Translator1Firstname & " " & Translator1Surname & ")")) &If (IsEmpty (Author1Surname & Editor1Surname & Translator1Surname) ; "" ; ": ") &If (IsEmpty (chapter_in_book) ; "" ; "" & chapter_in_book & "" & ", in ") &If (IsEmpty (BookTitle) ; "" ; "" & TextStyleAdd (BookTitle ; Italic) & "")&If (IsEmpty (no_of_volumes) ; "" ; " " & no_of_volumes & " vols.") & If (IsEmpty (series_title) ; "" ; ", " & series_title) &If (IsEmpty (vol_no_in_series) ; "" ; " Vol. " & vol_no_in_series & ", ") &If (IsEmpty (Editor1Surname); "" ; " edited by ") &If (IsEmpty (Editor1Surname) ; "" ; Editor1Firstname & " " & Editor1Surname) &If (IsEmpty(Editor4Surname); If (IsEmpty (Editor3Surname); If (IsEmpty(Editor2Surname); "" ; " and " & Editor2Firstname & " " & Editor2Surname) ; " " &Editor2Firstname & " " & Editor2Surname & " and " & Editor3Firstname & " " &Editor3Surname) ; " " & Editor2Firstname & " " & Editor2Surname &" " &Editor3Firstname & " " & Editor3Surname & " and " & Editor4Firstname & " " &Editor4Surname) & " (" & If (IsEmpty (PlacePublication) ; "" ; PlacePublication &If (IsEmpty (Publisher) ; ", " ; ": ") &If (IsEmpty (Publisher) ; "" ; Publisher & ", ")) & If (IsEmpty (Year) ; "n.d." ;Year) &")" & If (IsEmpty (Page) ; "" ; ", " & Page) &If (IsEmpty (notes) ; "" ; " [" & notes & "]") ;If (IsEmpty (Author1Surname) ; "" ; Author1Surname & ", " & Author1Firstname) &If (IsEmpty (Author2Surname) ; "" ; "; " & Author2Surname & ", " &Author2Firstname) &If (IsEmpty (Author3Surname) ; "" ; "; " & Author3Surname & ", " &Author3Firstname) &If (IsEmpty (Editor1Surname) ; "" ; Editor1Surname & ", " & Editor1Firstname) &If (IsEmpty (Editor2Surname); "" ; "; " & Editor2Surname & ", " & Editor2Firstname) &If (IsEmpty (Editor3Surname); "" ; "; " & Editor3Surname & ", " & Editor3Firstname) &If (IsEmpty (Editor1Surname); "" ;If (IsEmpty (Editor2Surname); " (ed.)" ; " (eds.)")) &//translator info//If (IsEmpty (Translator1Surname) ; "" ;If (IsEmpty (Author1Surname & Editor1Surname) ; "(trans. " & Translator1Firstname& " " &*

		Translator I Surname & ")" ;" (trans. " & Translator I Firstname & " " & Translator I Surname & ")")) &If (IsEmpty (Author I Surname & Editor I Surname & Translator I Surname) ; "" ; ": ") &If (IsEmpty (omit_Chapter_from_calculation) ;If (IsEmpty (chapter_in_book) ; "" ; """ & chapter_in_book & """ & ", in ") ; "in ") &If (IsEmpty (BookTitle) ; "" ;"" & TextStyleAdd (BookTitle ; Italic) & "") &If (IsEmpty (no_of_volumes) ; "" ; " " & no_of_volumes & " vols.") &If (IsEmpty (series_title) ; "" ; ", " & series_title) &If (IsEmpty (vol_no_in_series) ; "" ; " Vol. " & vol_no_in_series) &If (IsEmpty (ArticleTitle) ; "" ; """ & TextFormatRemove (ArticleTitle) & "" ") &If (IsEmpty (Journal); "" ; "" & TextStyleAdd (Journal ; Italic) & "" & If(IsEmpty (Vol) ; "" ; " ") & Vol) &If (IsEmpty (dissertation) ; "" ; "" & TextStyleAdd (dissertation ; Italic) &", " & degree & ", " & university) &If (IsEmpty (omit_Year_from_calculation) ; " (" ; "") &If (IsEmpty (PlacePublication) ; "" ; PlacePublication &If (IsEmpty (Publisher) ; "" ; ": ")) &If (IsEmpty (Publisher) ; "" ; Publisher) &If (IsEmpty (Publisher) and IsEmpty (PlacePublication) ; "" ; ", ") &If (IsEmpty (omit_Year_from_calculation) ; If (IsEmpty (Year) ; "n.d." ;Year) &")" ; "") &If (IsEmpty (Page) ; "" ; ", " & Page) &If (IsEmpty (notes) ; "" ; " [" & notes & "]")
InformationSource	Text	
Journal	Text	
logged_in_as	Calculation	(Text) from Bibliography, = *Get (AccountName) modification dateText Auto-enter: "Modification Timestamp (Date and Time)"*
no_of_volumes	Text	
notes	Text	
numberFound	Calculation	(Number) Unstored, from Bibliography, = *Get (FoundCount)*
omit_Chapter_from_calculation	Text	
omit_Year_from_calculation	Text	
Page	Text	
PlacePublication	Text	
Publisher	Text	
selfReference	Number	
series_title	Text	
university	Text	
Vol	Text	
vol_no_in_series	Text	
Year	Text	

BibliographyPage_IS

The table that links the bibliography to the **Source** *table. This table, as well as providing that connection, contains information about specific page numbers within an article or book that refer to the MS to which it is connected; in addition further information such as whether images are reproduced, what type of information the article might contain, and whether it includes editions of the music, is entered in the* CCM_info *field. This table also allows the user to define a bibliographical item as being of primary importance in describing the source manuscript, and the field* Marked_up_bibliography *is the one that appears in the bibliography list on the website either in bold or normal font, depending on whether the item has been marked as important by the database editor.*

Field Name	Field Type	Formula / Entry Option
BiblAbbrev	Text	
bibliographyKey	Text	Lookup: When an entry in "BibliographyPage_IS::BiblAbbrev" is made, copy the first matching value from "Bibliography::bibliographyKey" If no match: "Do not copy" don't copy "Bibliography::bibliographyKey" if empty
bibliographypage_IS_KEY	Number	Serial Number on creation with Current Value: "25071" Increment: "1", Required value, Unique values only, Only allow values of type: "Numeric Only"

CCM_info	Text	
editing_history	Calculation	(Text) from BibliographyPage_IS, = *If (IsEmpty (logged_in_as) ; editing_history ; Get (AccountName) & " " & modification date & ¶ & editing_history)*
fullBibliographyText	Text	Lookup: When an entry in "BibliographyPage_IS::BiblAbbrev" is made, copy the first matching value from "Bibliography::FullText_calculated" If no match: "Do not copy" don't copy "Bibliography::FullText_calculated" if empty
logged_in_as	Calculation	(Text) from BibliographyPage_IS, = *Get (AccountName)*
Marked_up_bibliography	Calculation	(Text) Unstored, from BibliographyPage_IS, = *If (IsEmpty (primary_study) ; Bibliography::FullText_calculated & If (IsEmpty (Page) ; "" ; ", " & Page) & If (IsEmpty (CCM_info) ; "" ;TextFont (" [CCM descriptor(s): " & CCM_info & "]" ; "Times")) ; "" & Bibliography::FullText_calculated & If (IsEmpty (Page) ; "" ; ", " & Page) & If (IsEmpty (CCM_info) ; "" ;TextFont (" [CCM descriptor(s): " & CCM_info & "]" ; "Times")) & "")*
modification date	Text	Auto-enter: "Modification Timestamp (Date and Time)"
Page	Text	
primary_study	Text	
sortorder	Number	
sourceKey	Text	

BiblItem_IS

LInks the **Item** *table to the Bibliography.*

Field Name	Field Type	Formula / Entry Option
biblAbbrev	Text	
biblItem_IS_KEY	Number	Serial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
bibliographyKey	Number	Lookup: When an entry in "BiblItem_IS::biblAbbrev" is made, copy the first matching value from "BibliographyItem::bibliographyKey" If no match: "Do not copy" don't copy "BibliographyItem::bibliographyKey" if empty
itemKey	Number	Lookup: When an entry in "BiblItem_IS::itemKey" is made, copy the first matching value from "Item::itemKey" If no match: "Do not copy" don't copy "Item::itemKey" if empty
notes	Text	

Catalogue_Pictures

This table serves one function only: to tell the SQL database for the webapp what the order of page images should be for the printed catalogue page images.

Field Name	Field Type	Formula / Entry Option
Cat_imagefilename	Text	
keyNo_order	Number	Serial Number on creation with Current Value: "9488" Increment: "1", Allow user to override validation, Unique values only, Only allow values of type: "Numeric Only"

CatalogueIndexes

This table does not appear in the online resource: it is used to allow easy navigation through the printed catalogue images and is for internal use for those populating the back-end database. It may be implemented online at a later date for RISM

Field Name	Field Type	Formula / Entry Option
A_Z	Text	

archiveKey	Number	
catalogueindexesKey	Number	Serial Number on creation with Current Value: "1196" Increment: "1", Allow user to override validation, Required value, Unique values only
CCMvol1	Text	
CCMvol2	Text	
CCMvol3	Text	
CCMvol4	Text	
countryKey	Number	
RISM_BIV_1	Text	
RISM_BIV_2	Text	
RISM_BIV_3	Text	
RISM_BIV_4	Text	
RISM_BIV_5	Text	
RISM_BIV_supplement	Text	

Composerlink_IS

This table links compositions to composers. In the period covered by the sources in this dataset it is not uncommon to find works ascribed to more than one composer, or for an ascription to be uncertain. The connection between the two tables is made using this intersection set, which is also used to define whether the connection is only an uncertain one, and allows for a note to be added to the attribution. Many of the fields in this database are redundant copies of fields from the connected tables to facilitate display in the working database.

Field Name	Field Type	Formula / Entry Option
attribution_uncertain	Text	
composerKey	Number	
composerlink_IS_KEY	Number	Serial Number on creation with Current Value: "21691" Increment: "1", Required value, Unique values only, Only allow values of type: "Numeric Only"
Composername	Calculation	(Text) Unstored, = *If (IsEmpty (Composers::firstname) ; "" ; Composers::firstname & " ") & If (IsEmpty (Composers::lastname) ; "" ; Composers::lastname)*
composerNameOriginal	Text	
compositionKey	Number	
date	Text	Lookup: When an entry in "Composerlink_IS::composerKey" is made, copy the first matching value from "Composers::dates_public" If no match: "Do not copy" don't copy "Composers::dates_public" if empty
firstname	Text	Lookup: When an entry in "Composerlink_IS::composerKey" is made, copy the first matching value from "Composers::firstname" If no match: "Do not copy" don't copy "Composers::firstname" if empty
itemKey	Text	Lookup: When an entry in "Composerlink_IS::compositionKey" is made, copy the first matching value from "Item::itemKey" If no match: "Do not copy" don't copy "Item::itemKey" if empty
notes_attribution	Text	
sourceKey	Text	Lookup: When an entry in "Composerlink_IS::compositionKey" is made, copy the first matching value from "Item::sourceKey" If no match: "Do not copy" don't copy "Item::sourceKey" if empty

Composers

The master list of composers whose works appear in the manuscripts of the dataset. Composers from this period often had variant spellings of their name, or went under numerous aliases when linked to their compositions. Often we know so little of their biography that we can only give approximate dates for their activity, and information varies from the specific to the vague. The content of this database is largely not available online—it is reasonably easy for a user to find out more about a composer from e.g. GroveOnline, and this resource is not intended to provide a composer biography for every composer in the period. However for administrative purposes it is useful to have biographical information as a way of assisting us in working out which composer

might be intended by an ascription that is not clear. The table is linked to the **Composition** *table via an intersection set that allows us to define whether an ascription is certain or dubious, and is also connected to the* **Bibliography** *table via intersection set.*

Field Name	Field Type	Formula / Entry Option
composerComplete	Calculation	(Text) from Composers, = *lastname & If (IsEmpty (firstname) ; "" ; ", " & firstname) & If (IsEmpty (variantspellings) ; "" ; " (" & variantspellings & ")")*
composerComplete_reverse		Calculation(Text) Indexed, from Composers, = *TextFormatRemove (lastname & If (IsEmpty (firstname) ; "" ; ", " & firstname) & If (IsEmpty(variantspellings) ; "" ; " (" & variantspellings & ")"))*
composerKey	Number	Serial Number on creation with Current Value: "1082" Increment: "1", Required value, Unique values only, Only allow values of type: "Numeric Only"
date_earliest	Text	
date_floruit_earliest	Text	
date_floruit_latest	Text	
date_latest	Text	
dates_public	Text	
firstname	Text	
g_Key	Number	Indexed, Auto-enter: "99", Allow user to override validation, Required value, Only allowvalues of type: "Numeric Only"
info	Text	
info_source	Text	
lastname	Text	
variantspellings	Text	

Composition

Most pieces of music surviving from this period are recorded in more than one manuscript, some in many manuscripts. In the case of partbooks the same composition appears in each book, but each book has a different voice part. It is necessary therefore to create a master list of compositions, which uses a standardised form of title spelling, to which all instances of this composition, or individual voice parts belonging to it, can be linked. The composition is therefore not a 'real' entity but the intellectualised version of any number of individual items in a variety of manuscript (and printed) sources. In many ways this is what is represented by an editor's published modern edition of a work, which usually collates all the individual instances into a single idealised version of the work, which does not represent any original reading or single source for the work. The composition links individual items in manuscripts with the **Composers**, *with the* **Composition_cycle** *to which individual items may belong, to the genre of the work (***alGenre***), to the* **Composition_model** *(e.g. a mass may be based on a motet which will also be listed in the* **Composition** *table, or a work may use a plainchant for its basis or tenor part), and to* **Bibliography** *relation to that particular composition. The number of voices in a work can vary, sometimes by design and sometimes because the piece is preserved incomplete. The* **Item** *table show the number of voices shown in that instance of a composition, but this table shows the maximum number of voices.*

Field Name	Field Type	Formula / Entry Option
attribution_authority	Text	
composerAttribution	Text	Lookup: When an entry in "Composition::compositionKey" is made, copy the first matching value from "Item::composerOriginal" If no match: "Do not copy" don't copy "Item::composerOriginal" if empty
composition_name	Text	Lookup: When an entry in "Composition::compositionKey" is made, copy the first matching value from "Item::composition_name" If no match: "Do not copy" don't copy "Item::composition_name" if empty
compositionKey	Number	Serial Number on creation with Current Value: "51254" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
compositionModelKey	Number	Auto-enter: "0"
isorhythmic	Text	
max_number_of_voices	Number	
notes_concordances	Text	

TITLE	Text	

Composition_cycle

This table allows us to create a governing cycle which a set of compositions may belong to. Cycles range from groupings of mass movements (which may not be copied consecutively in a MS) to motet cycles created by a publisher. It is also used to link 'pars' items in motets or other works that are split into internal sections that may sometimes appear as individual works in different sources.

Field Name	Field Type	Formula / Entry Option
composer	Text	Lookup: When an entry in "Composition_cycle::composerKey" is made, copy the first matching value from "CompCycle_Composer::lastname" If no match: "Do not copy" don't copy "CompCycle_Composer::lastname" if empty
composerKey	Number	Lookup: When an entry in "Composition_cycle::composition_cycleKey" is made, copy the first matching value from "Composers::composerKey" If no match: "Do not copy" don't copy "Composers::composerKey" if empty
composition_cycleKey	Number	Serial Number on creation with Current Value: "24" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
cycleTypeKey	Number	
title	Text	
title_composer	Calculation	(Text) = *title* & " - " & *composer*
title_model_compositionKey		Number

CompositionGenre_IS

Intersection set between **Composition** *and* **Genre**. *This allows us to designate more than one genre per composition. the term 'genre' is used quite loosely, allowing us to designate both large and small groups of works to which a composition belongs. For example, a work entitled 'Ave Regina' is designated both 'motet' and 'ave regina', since 'ave regina' is an antiphon text set very widely and by many composers. Similarly, works such as mass movements, would be designated both 'kyrie' and 'mass ordinary'.*

Field Name	Field Type	Formula / Entry Option
compositionGenre_IS_KEY	Number	Serial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
compositionKey	Number	
genre	Calculation	(Text) Unstored, = *alGenre::Genre*
genreKey	Number	

Cycle_item_IS

Musical works (compositions) can belong to cycles, but can also belong to more than one cycle, so this intersection set connects individual items to a defined cycle. The table name is slightly misleading, since it does not connect cycles to individual items in the **Item** *table, but rather to the intellectual construct, 'composition', and therefore connects the two tables* **Cycle** *and* **Composition**. *Cycles take a variety of forms: this may be a set of mass movements that together form a complete mass setting, but this may also refer to individual 'pars' within a motet which have been recorded as individual compositions and items since one pars may appear in a source as a work in its own right, divorced from its cycle companions. Complete motets also may belong to cycles of motets. The cycle therefore is a broadly defined connection between two or more musical works or compositions.*

Field Name	Field Type	Formula / Entry Option
composition_cycleKey	Number	
compositionKey	Number	Lookup: When an entry in "Cycle_item_IS::compositionKey" is made, copy the first matching value from "Composition::compositionKey" If no match: "Do not copy" don't copy "Composition::compositionKey" if empty
cycle_item_IS_KEY	Number	Serial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
order_in_cycle	Number	

position_title	Text	

Image

The **Image** *table is one of the primary tables in the database: it lists the individual pages, flyleaves, covers and any other items that would have been photographed while making a record of a mansucript. Hierarchically it would come immediately 'below' the* **Source** *table, since each record would belong to a single source. This table does not in fact simply list the images taken when photographing a MS, but instead lists the pages/folios, endpapers, boards, covers etc. Therefore if more than one image was taken of a page there would be only one record in this table, but the additional images (such as UV, IR or detail shots) are recorded in the* **SecondaryImages** *table. This includes images that have been digitally restored or otherwise manipulated. The* **Item** *and* **Image** *tables are linked via an intersection set (***ItemImage_IS***). The content includes information about the storage of images, but mainly allows the online delivery system to find the correct image in relation to a folio number.*

Field Name	Field Type	Formula / Entry Option
archive	Text	Lookup: When an entry in "Image::sourceKey" is made, copy the first matching value from "Source::archive" If no match: "Do not copy" don't copy "Source::archive" if empty
archiveddetailfilename	Text	
archivedfilename	Text	
archivedUVfilename	Text	
archivedVRfilename	Text	
archivedwatermarkfilename		Text
availwebsite	Text	Lookup: When an entry in "Image::sourceKey" is made, copy the first matching value from "Archive::availableOnWebsite" If no match: "Do not copy" don't copy "Archive::availableOnWebsite" if empty
bibliography	Text	
brightness	Text	
captureconditions	Text	
capturedevice	Text	
city	Text	Lookup: When an entry in "Image::sourceKey" is made, copy the first matching value from "Source::city" If no match: "Do not copy" don't copy "Source::city" if empty
completesourcetitle	Calculation	(Text) Unstored, from Image, = *city* & " " & *archive* & " " & *source* & " *f.* " & *folio* & " © " & *Archive::copyrightholder*
copyrightstatement	Text	Lookup: When an entry in "Image::sourceKey" is made, copy the first matching value from "Archive::copyrightholder" If no match: "Do not copy" don't copy "Archive::copyrightholder" if empty
currentmeasurements	Text	
dateCopied	Text	
dateCreated	Date	Auto-enter: "Creation Date"
dateModified	Date	Auto-enter: "Modification Date"
detail	Text	
detailfilename	Text	
digitised	Text	
EECMno	Text	
existingimages	Text	
filename	Text	
folio	Text	
folio_alt	Text	
imageKey	Number	Serial Number on creation with Current Value: "26789" Increment: "1", Required value, Unique values only, Only allow values of type: "Numeric Only"
imagetype	Text	
incipitimagefilename	Text	Lookup: When an entry in "<unknown>" is made, copy the first matching value from "<Table Missing>" If no match: "Do not copy" don't copy

		"<Table Missing>" if empty - something to do with ALTINCIPIT
notes	Text	
orderNo	Number	Serial Number on creation with Current Value: "204" Increment: "1", Allow user to override validation, Required value, Only allow values of type: "Numeric Only"
RISMimagefilename	Text	Lookup: When an entry in "Image::sourceKey" is made, copy the first matching value from "Source::RISMimagefilename1" If no match: "Do not copy" don't copy "Source::RISMimagefilename1" if empty
RISMimagefilename2	Text	
source	Text	Lookup: When an entry in "Image::sourceKey" is made, copy the first matching value from "Source::shelfMark alone" If no match: "Do not copy" don't copy "Source::shelfMark alone" if empty
sourceKey	Number	Lookup: When an entry in "Image::sourceKey" is made, copy the first matching value from "Source::sourceKey" If no match: "Do not copy" don't copy "Source::sourceKey" if empty
surfaceMaterial	Text	
TSM	Text	
watermark	Text	

Item

The **Item** *table is one of the primary tables in the database: it lists the individual items within a mansucript (i.e. the inventory), and hierarchically would come immediately 'below' the* **Source** *table. The items represent the individual and specific appearance of a work in a manuscript, not the intellectual concept of the work, so each entry is unique. Each work in a manuscript is described as an inventory ITEM. However we do not list non-musical contents of manuscripts, or at least not consistently. Occasionally a source might contain blocks of monophonic chant interleaved with polyphonic items. In this case the first 'item' might say 'items 1-37 mass proper chants' and the next item would refer to the first polyphonic work in the manuscript. This table has one of the widest set of interconnections to other tables, the most important one being to the* **Composition** *table, in which each instance of a work (entry in the item table) is linked to the intellectual concept of a 'composition' - the idealised form of the work to which all individual instances of that work are linked. The* **Item** *and* **Image** *tables are linked via an intersection set (***ItemImage_IS***).*

Field Name	Field Type	Formula / Entry Option
_PracticeDBwarning	Calculation	(Text) Unstored, from Archive, = TextColor (If (Right (Get (FileName) ; 7) = "RUBBISH" ;"YOU ARE IN THE PRACTICE DATABASE" ;""); RGB (0 ; 0 ; 0))
AbbrevPosn	Calculation	(Text) from Item, = *RISMabbrev* & " " & *PositioninMS*
alt_numbering	Text	
altincipitfilename	Text	
altincipititemKey	Number	Lookup: When an entry in "Item::itemKey" is made, copy the first matching value from "altincipit by itemKey::altitemKey" If no match: "Do not copy" don't copy "altincipit by itemKey::altitemKey" if empty
blueDog	Text	
catalogDesignation	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::24. Catalog designation" If no match: "Do not copy" don't copy "Motetten-Datenbank::24. Catalog designation" if empty
commentsonhands	Text	
complete Library Info	Calculation	(Text) Unstored, from Item, = *Source::Complete Library info*
completeText_original	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::36. CompleteText - Original" If no match: "Do not copy" don't copy "Motetten-Datenbank::36. Complete Text - Original" if empty
completeText_standard	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::37. CompleteText - Standard" If no match: "Do not copy" don't copy "Motetten-Datenbank::37.

		Complete Text - Standard" if empty
composerOriginal	Text	
composerStandard	Text	
composition_gathered	Calculation	(Text) from Item, = *Substitute (composition_gathered ; ["¶" ; " / "] ; ["[untexted] / " ; ""] ; [" / [untexted]" ; ""] ; [" " ; ""]) composition_name Calculation(Text) Unstored, from Item, = If (IsEmpty (useMotet_DB_title) ; If (IsEmpty (useGenre) ; If (IsEmpty (composition_gathered) ; Motet_incipit_standard ; composition_gathered) ; alGenre::Genre) ; Motet_incipit_standard)*
compositionKey	Number	
concordances	Text	
dateComposed	Text	
dateCopied	Text	
DIAMMlink	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::DIAMMlink" If no match: "Do not copy" don't copy "Motetten-Datenbank::DIAMMlink" if empty
EarpDesignation	Calculation	(Text) Unstored, from Item, = *Item to Source::EarpDesignation*
editing_history	Calculation	(Text) from Item, = *If (IsEmpty (logged_in_as) ; editing_history ; Get (AccountName) & " " & modification date & ¶ & editing_history) from Item, = If (IsEmpty (logged_in_as) ; editing_history ; Get (AccountName) & " " & modification date & ¶ & editing_history)*
edition	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::31. Edition" If no match: "Do not copy" don't copy "Motetten-Datenbank::31. Edition" if empty
foliationPagination	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::26. Foliation/Pagination" If no match: "Do not copy" don't copy "Motetten-Datenbank::26. Foliation/Pagination" if empty
folio_end	Text	
folio_end_alt	Text	
folio_end_serial	Calculation	(Number) Unstored, from Item, = *folioend to folio::sourceKey & folioend to folio::orderNo*
folio_end split	Calculation	(Number) from Item, = *If (PatternCount (Folios ; "-") ; Middle (Folios ; (Position (Folios ; "-" ; 1 ; 1) +1) ; 999) ; Substitute (Folios ; "p. " ; ""))*
folio_start	Text	
folio_start_alt	Text	
folio_start_serial	Calculation	(Number) Unstored, from Item, = *foliostart to folio::sourceKey & foliostart to folio::orderNo*
folio_start spilt	Calculation	(Text) from Item, = *If (PatternCount (Folios ; "-") ; Middle (Folios ; 1 ; (Position (Folios ; "-" ; 1 ; 1)) -1) ; Folios)*
Folios	Text	
g_composerkey	Calculation	(Number) from Item, = "99"
g_Key	Number	Indexed, Auto-enter: "99", Allow user to override validation, Required value, Only allowvalues of type: "Numeric Only"
genre	Text	Lookup: When an entry in "Item::compositionKey" is made, copy the first matching value from "alGenre::Genre" If no match: "Do not copy" don't copy "alGenre::Genre" if empty
genreKey	Number	Auto-enter: "0"
Global	Number	Global field used in relationship between Item and Text to show all records
incipitfilename	Text	Auto-enter Calculation: from Item, = *If (IsEmpty (incipitfilename) ; Item to Source::RISMimagefilename1 ; "")*

item_gathered	Calculation	(Text) Unstored, from Item, = *If (IsEmpty (PositioninMS) ; "" ; PositioninMS & ". ") & If (IsEmpty (folio_start) ; "" ; folio_start & If (folio_start = folio_end ; ": " ; If (IsEmpty (folio_end) ; ": " ; "-" & folio_end & ": ")))& If (IsEmpty (useMotet_DB_title) ; composition_name ; Substitute (Motet_incipit_standard ; " - " ; " / ")) & If (IsEmpty (composerStandard) ; If (IsEmpty (Composerlink_IS::Composername) ; "" ; " — " & Composerlink_IS::Composername) ; " — " & composerStandard)*
itemKey	Number	Serial Number on creation with Current Value: "61653" Increment: "1", Always validate, Required value, Unique values only, Only allow values of type: "Numeric Only"
Last edited by	Text	
layout	Text	
local_attribution	Text	
logged_in_as	Calculation	(Text) from Item, = *Get (AccountName)*
modification date	Text	Auto-enter: "Modification Timestamp (Date and Time)"
MOTET	Text	
Motet_incipit_standard	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::32a. Motet incipit - Standard" If no match: "Do not copy"
MOTET_order	Number	Auto-enter Calculation: from Item, = *If (IsEmpty (${Motetten-Datenbank}:: ${25. Order in source}) ; "0" ; ${Motetten-Datenbank}::${25. Order in source}) IFnodremxeudla, A/ Eutnot-reyn Otepr tciaolnculation: from Item, = If (IsEmpty (${Motetten-Datenbank}:: ${25. Order in source}) ; "0" ; ${Motetten-Datenbank}::${25. Order in source})*
MOTETcatalog	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::24. Catalog designation" If no match: "Do not copy" don't copy "Motetten-Datenbank::24. Catalog designation" if empty
MotetComments	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::30. Motet comments" If no match: "Do not copy" don't copy "Motetten-Datenbank::30. Motet comments" if empty
MOTETincipitfilename	Text	
MOTETkey	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::motetKey" If no match: "Do not copy" don't copy "Motetten-Datenbank::motetKey" if empty
MOTETshelfmark	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::MOTET shelfmark" If no match: "Do not copy" don't copy "Motetten-Datenbank::MOTET shelfmark" if empty
MusicalGenre	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::27. Musical genre" If no match: "Do not copy" don't copy "Motetten-Datenbank::27. Musical genre" if empty
musicalIncipit_score	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::38. Musical incipit - Score" If no match: "Do not copy" don't copy "Motetten-Datenbank::38. Musical incipit - Score" if empty
notes	Text	
noVoices	Text	
null_value	Text	Global
numberGarbage_RISM5	Text	

orderNo	Number	Serial Number on creation with Current Value: "3281" Increment: "1", Allow user to override validation, Required value
pageNextRISM	Calculation	(Text) Unstored, from Item, = *Left (incipitfilename ; 2) & If (Length (incipitfilename) = 5 ; Case (Length (Right (incipitfilename ; 3) + 1) = 3 ; Right (incipitfilename ; 3) + 1 ; Length (Right (incipitfilename ; 3) + 1) = 2 ; "0" & Right (incipitfilename ; 3) + 1 ; Length (Right (incipitfilename ; 3) + 1) = 1 ; "00" & Right (incipitfilename ; 3) + 1 ; "Whoops - you've run out of pages in this volume !") ; Case (Length (Right (incipitfilename ; 4) + 1) = 4 ; Right (incipitfilename ; 4) + 1 ; Length (Right (incipitfilename ; 4) + 1) = 3 ; "0" & Right (incipitfilename ; 4) + 1 ; Length (Right (incipitfilename ; 4) + 1) = 2 ; "00" & Right (incipitfilename ; 4) + 1 ; Length (Right (incipitfilename ; 4) + 1) = 1 ; "000" & Right (incipitfilename ; 4) + 1 ; "Whoops - you've run out of pages in this volume !"))*
pagePreviousRISM	Calculation	(Text) Unstored, from Item, = *Left (incipitfilename ; 2) & If (Length (incipitfilename) = 5 ; Case (Length (Right (incipitfilename ; 3) - 1) = 3 ; Right (incipitfilename ; 3) - 1 ; Unstored, from Item, = Left (incipitfilename ; 2) & If (Length (incipitfilename) = 5 ; Case (Length (Right (incipitfilename ; 3) - 1) = 3 ; Right (incipitfilename ; 3) - 1 ; Length (Right (incipitfilename ; 3) - 1) = 2 ; "0" & Right (incipitfilename ; 3) - 1 ; Length (Right (incipitfilename ; 3) - 1) = 1 ; "00" & Right (incipitfilename ; 3) - 1 ; "Whoops - you've run out of pages in this volume !") ; Case (Length (Right (incipitfilename ; 4) - 1) = 4 ; Right (incipitfilename ; 4) - 1 ; Length (Right (incipitfilename ; 4) - 1) = 3 ; "0" & Right (incipitfilename ; 4) - 1 ; Length (Right (incipitfilename ; 4) - 1) = 2 ; "00" & Right (incipitfilename ; 4) - 1 ; Length (Right (incipitfilename ; 4) - 1) = 1 ; "000" & Right (incipitfilename ; 4) - 1 ; "Whoops - you've run out of pages in this volume !"))*
pars	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::32. Pars" If no match: "Do not copy" don't copy "Motetten-Datenbank::32. Pars" if empty
parsTitle	Text	
piecenumber	Text	
PositioninMS	Number	
PositiononPage	Text	
RISM5_text	Text	
RISMabbrev	Text	Lookup: When an entry in "Item::sourceKey" is made, copy the first matching value from "Item to Source::RISMabbrev" If no match: "Do not copy" don't copy "Item to Source::RISMabbrev" if empty
scribalhabit	Text	
Source	Text	Lookup: When an entry in "Item::sourceKey" is made, copy the first matching value from "Item to Source::sourceDisplayName" If no match: "Do not copy" don't copy "Item to Source::sourceDisplayName" if empty
sourceKey	Number	
StdTextSource	Text	
text_incipit_source	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::33.Text Incipit-Source" If no match: "Do not copy" don't copy "Motetten-Datenbank::33.T e x t Incipit-Source" if empty
text_incipit_standard	Text	Lookup: When an entry in "Item::MOTETkey" is made, copy the first matching value from "Motetten-Datenbank::34.Text Incipit-Standard" If no match: "Do not copy" don't copy "Motetten-Datenbank::34.T e x t Incipit-Standard" if empty
useGenre	Text	

useMotet_DB_title	Text

Item_Copyist_IS

Intersection set betwen **Item** *records and the* **Copyist** *table. Although the* **Source** *table is also linked to the* **Copyist** *table by an intersection set, this intersection allows us to indicate which of the individual items in a manuscript were the work of a particular copyist, since often MSS are the work of a number of compilers.*

Field Name	Field Type	Formula / Entry Option
attribution_uncertain	Text	
copyistKey	Number	
copyistTypeKey	Number	
item_Copyist_IS_KEY	Number	Serial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
itemKey	Number	

Item_Person_IS

Just as the **Source** *table links manuscripts to individuals or institutions via the* **Source_Person_IS**, *this intersection allows individual items to be linked to entries in the* **alPerson** *table.*

Field Name	Field Type	Formula / Entry Option
attribution_uncertain	Text	
itemKey	Number	
item_Person_IS_KEY	Number	Serial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
personKey	Number	
personRelationshipKey	Number	

ItemImage_IS

As its name suggests, this table connects the **Item** *and* **Image** *tables, generating a connection that allows us to see which images match which items. This intersection is necessary because there may be more than one item on a page, or an item may run for several pages.*

Field Name	Field Type	Formula / Entry Option
altincipitfilename	Text	Lookup: When an entry in "ItemImage_IS::itemKey" is made, copy the first matching value from "Item::altincipitfilename" If no match: "Do not copy" don't copy "Item::altincipitfilename" if empty
folio	Text	Lookup: When an entry in "ItemImage_IS::itemKey" is made, copy the first matching value from "Item::Folios" If no match: "Do not copy" don't copy "Item::Folios" if empty
imageKey	Text	
incipitfilename	Text	Lookup: When an entry in "ItemImage_IS::itemKey" is made, copy the first matching value from "Item::incipitfilename" If no match: "Do not copy" don't copy "Item::incipitfilename" if empty
itemimage_IS_KEY	Number	Serial No on creation with Current Value: "3782" Increment: "1", Required value, Unique values only, Only allow values of type: "Numeric Only"
itemKey	Text	
orderNo	Number	Serial Number on creation with Current Value: "1" Increment: "1"
sourceKey	Text	Lookup: When an entry in "ItemImage_IS::imageKey" is made, copy the first matching value from "Image::sourceKey" If no match: "Do not copy" don't copy "Image::sourceKey" if empty

Lost Motets

THIS TABLE DOES NOT APPEAR IN THE SQL DATABASE. This is a list of motets found in the MOTET database that don't ap-

pear to have an analog in our DB or existing inventories. They have therefore been removed from the DIAMM item table, and placed here rather than deleting them altogether.

Field Name	Field Type	Formula / Entry Option
motetKey	Number	
notes	Text	
title	Text	Lookup: When an entry in "Lost Motets::motetKey" is made, copy the first matching value from "Motetten-Datenbank::32a. Motet incipit - Standard" If no match: "Do not copy" don't copy "Motetten-Datenbank::32a. Motet incipit - Standard" if empty

Motetten-Datenbank

THIS TABLE DOES NOT APPEAR IN THE SQL DATABASE. This is the original table supplied from the MOTET database project and is linked to the Source and Item tables, which derive some data from this table via relationship and lookups. When this table is redundant it will be deleted from the database. It is kept in place for the moment in case any data has failed to copy over correctly. Numbered fields are those originally in the MOTET database (which was not relational), unnumbered fields have been added for data transfer purposes. Most of these fields turned out to be empty

Field Name	Field Type	Formula / Entry Option
01. SourceName	Text	
02. RISM abbrev	Text	
03. CC Sigla	Text	
04. Startdate	Text	
05. Enddate	Text	
06. Intdate	Text	
07. Date-Comments	Text	
08. Dedicatee	Text	
09. Dedicator	Text	
10. Dedication-text	Text	
11. Liminary Text	Text	
12. Scribe/Publisher	Text	
13. Editor	Text	
14. Establishment/Patron	Text	
15. Source-City	Text	
16. Source-Country	Text	
17. Source-Region	Text	
18. Source-Citation	Text	
19. Contributor	Text	Auto-enter value from last visited record
20. Source-Comments	Text	
21. Source type	Text	
22. Composer-original	Text	
22a. Conflicting attributions	Text	
23. Composer-standard	Text	
24. Catalog designation	Text	
25. Order in source	Number	
26. Foliation/Pagination	Text	
27. Musical genre	Text	
28. Text type	Text	
29. Motet-source comments	Text	
30. Motet comments	Text	
31. Edition	Text	
32. Pars	Text	
32a. Motet incipit - Standard	Text	
33. Text Incipit-Source	Text	
34. Text Incipit-Standard	Text	
35. Voices	Text	

36. Complete Text - Original	Text	
37. Complete Text - Standard	Text	
38. Musical incipit - Score	Container	
DIAMMlink	Calculation	(Text) Unstored, from Motetten-Datenbank, = *MOTET to source::RISMabbrev* & " " & *${25. Order in source}*
DIAMMsourceKey	Number	
editing_history	Calculation	(Text) from Motetten-Datenbank, = *If (IsEmpty (logged_in_as) ; editing_history ; Get (AccountName)* & " " & *modification date* & ¶ & *editing_history)*
incipitFilename	Text	
itemKey	Number	Lookup: When an entry in "Motetten-Datenbank::motetKey" is made, copy the first matching value from "Item::itemKey" If no match: "Do not copy" don't copy "Item::itemKey" if empty
list fields	Calculation	(Text) Unstored, from Motetten-Datenbank, = *FieldNames ("DIAMM" ; "Thomas")*
logged_in_as	Calculation	(Text) from Motetten-Datenbank, = *Get (AccountName)*
modification date	Text	Auto-enter: "Modification Timestamp (Date and Time)"
MOTET shelfmark	Text	
motetKey	Number	Serial Number on creation with Current Value: "15173" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"

NotationSource_IS

This table links the **Source** *table to the authority list of music notation types* **alNotation**.

Field Name	Field Type	Formula / Entry Option
notationSource_IS_KEY	Number	Serial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
notationTypeKey	Number	
sourceKey	Number	

RISM_CCMlink

Not used in SQL. The table lists countries and cities with the relevant page number in both RISM and CCM where the entry for that country or city starts.

Field Name	Field Type	Formula / Entry Option
RISM_CCMimagefilename		Text
RISM_CCMlinkKey	Number	Serial Number on creation with Current Value: "770" Increment: "1", Allow user to override validation, Unique values only, Only allow values of type: "Numeric Only"
RISMpicturesKey	Text	Lookup: When an entry in "RISM_CCMlink::RISM_CCMimagefilename" is made, copy the first matching value from "RISM_CCMlink to RISMpictures::<Field Missing>" If no match: "Do not copy" don't copy "RISM_CCMlink to RISMpictures::<Field Missing>" if empty
SourceKey	Number	Lookup: When an entry in "RISM_CCMlink::SourceKey" is made, copy the first matching value from "Source::sourceKey" If no match: "Do not copy" don't copy "Source::sourceKey" if empty

SecondaryImages

Secondary images are different versions of primary images; it allows us to link UV, IR, detail and enhanced images to the primary image of a folio, and shows on the web interface in the image-viewer tool called 'Alternative images'. It connects only to the **Image** *table.*

Field Name	Field Type	Formula / Entry Option

archivedfilename	Text	
caption	Text	Allow user to override validation, Maximum number of characters = 255
datemodified	Date	Auto-enter: "Modification Date"
filename	Text	
ImageKey	Number	
imagetype	Text	
notes	Text	
SecondaryImagesKey	Number	Serial Number on creation with Current Value: "1350" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"

Set

Manuscripts or fragments may belong to sets - e.g. a set of partbooks where the individual members are listed as individual sources in the **Source** *table. The* **Set** *table is placed hierarchically above the* **Source** *table, but not between* **Archive** *and* **Source**, *where it might be expected to reside, since this would have involved allocating every manuscript to a set, even if this was a set of only one. Also, since sources may belong to more than one set (e.g. partbooks, and Alamire books), the connection between* **Set** *and* **Source** *has to be made through an intersection set (***SourceSet_IS***), which would cause further redundancy if every source was included. Therefore this table is tangentially 'above'* **Source**, *and is only linked to those sources that need to be allocated to one or more sets. There are various types of set, and these are defined in the authority list* **alSetType**.

Field Name	Field Type	Formula / Entry Option
bibliography	Text	
clusterShelfMark	Text	
description	Text	
setKey	Number	Serial Number on creation with Current Value: "251" Increment: "1", Allow user to override validation, Required value, Unique values only
setTypeKey	Number	

Source

This is a PRIME table, and is the most heavily populated of the dataset. At the moment all of the online data revolves around this table and its content, the master list of manuscripts.

Field Name	Field Type	Formula / Entry Option
_PracticeDBwarning	Calculation	(Text) Unstored, from Archive, = TextColor (If (Right (Get (FileName) ; 7) = "RUBBISH" ;"YOU ARE IN THE PRACTICE DATABASE" ;""); RGB (0 ; 0 ; 0))
abbrevs_checked	Text	
alt_numbering_source	Text	
altRISMabbrev	Text	
archive	Text	Lookup: When an entry in "Source::archiveKey" is made, copy the first matching value from "Archive::archiveName" If no match: "Do not copy" don't copy "Archive::archiveName" if empty
archiveKey	Number	
authority	Text	
bibliography	Calculation	(Text) from Source, = " " & *Substitute (If (IsEmpty (bibliography_gathered) ; bibliography_orig_abbrevs ; bibliography_gathered) ; "¶" ; " ") & " " & If (IsEmpty (CCMimagefilename1_delete) and IsEmpty (CCMimagefilename2_delete) and IsEmpty (RISMimagefilename1) and IsEmpty (RISMimagefilename2) and IsEmpty (RISMimagefilename3) ; " This manuscript is not listed in RISM or CCM. Please notify the Project Manager by e-mail of any information to supplement that displayed here. All contributions to information about manuscripts are acknowledged. " ; " Bibliographical information displayed has been taken from RISM and/or the Census Catalogue and may*

		therefore be out of date, though any more recent items notified to us have been included. We would be very grateful to be notified by e-mail of new items for inclusion in the bibliography lists and general information for any manuscript. All contributions to information about manuscripts are acknowledged. </p>")
BIBLIOGRAPHY_DONE	Text	
bibliography_gathered	Text	
bibliography_orig_abbrevs		Text
bibliographyParas	Calculation	(Text) from Source, =*TextSize (TextFont (Substitute (bibliography_orig_abbrevs ; ["; 11" ; "; II"] ; ["¶¶" ; "¶"]; ["¶" ; "¶¶"] ; ["; I" ; "§"] ; ["; LX" ; "goombas"] ; ["; X" ; "†"]; ["; V" ; "$"] ; ["; Appendix" ; "%"]; ["; " ; "¶"] ; ["§" ; "; I"]; ["†" ; "; X"] ; ["$" ; "; V"] ; ["goombas" ; "; LX"] ; ["%" ; "; Appendix"]) ; "Times New Roman"); 11)*
block	Text	
Blue Dogs	Text	
CCMabbrev	Text	Lookup: When an entry in "Source::sourceKey" is made, copy the first matching value from "Motetten-Datenbank::03. CC Sigla" If no match: "Do not copy" don't copy "Motetten-Datenbank::03. CC Sigla" if empty
CCMimagefilename1	Text	
CCMimagefilename1_delete		Text
CCMimagefilename2	Text	
CCMimagefilename2_delete		Text
CCMimagefilename3	Text	
CCMimagefilename3_delete		Text
CCMpresence	Calculation	(Text) from Source, = *If (IsEmpty (CCMimagefilename1_delete) ; "" ; "CCM")*
CCMreset	Text	Auto-enter Calculation: from Source, = *CCMimagefilename1_delete*
CCMreset_2	Text	Auto-enter Calculation: from Source, = *CCMimagefilename2_delete*
city	Text	Lookup: When an entry in "Source::archiveKey" is made, copy the first matching value from "Archive::city" If no match: "Do not copy" don't copy "Archive::city" if empty
Complete Library info	Calculation	(Text) Unstored, from Source, = *If (IsEmpty (moved) ;TextSize (TextFont (TextStyleRemove (TextStyleRemove (TextStyleRemove (TextColor (Archive::siglum ; RGB (18 ; 118 ; 18)) & " " &Archive::city & " " &Archive::archiveName & "," &shelfMark_alone & " " &If (IsEmpty (sourceName) ; "" ; " ('" & sourceName & "') ") &If (IsEmpty (olim) ; "" ; " [olim: " & olim & "] ") &/*TextColor (If (IsEmpty (dateOfSource) ; " NO DATE GIVEN" ; "- " & dateOfSource) ; RGB (18 ; 118 ; 18)) & */TextColor (If (IsEmpty (CCMabbrev) ; "" ; "; " & CCMabbrev) &If (IsEmpty (RISMabbrev) ; "" ; "; " & RISMabbrev) &If (IsEmpty (altRISMabbrev) ; "" ; "; " & altRISMabbrev) ; RGB (196 ; 5 ; 5)) ; Bold) ; Italic) ; "Arial") ; 13) ;TextColor ("This MS has moved: " ; RGB (196 ; 5 ; 5)) &//LOTS OF TEXT//TextSize (TextFont (TextStyleRemove (TextStyleRemove (TextColor (Archive::siglum ; RGB (18 ; 118 ; 18)) & " " &Archive::city & " " &Archive::archiveName & "," &shelfMark_alone & " " &If (IsEmpty (sourceName) ; "" ; " ('" & sourceName & "') ") &If (IsEmpty (olim) ; "" ; " [olim: " & olim & "] ") &/*TextColor (If (IsEmpty (dateOfSource) ; " NO DATE GIVEN" ; "- " & dateOfSource) ; RGB (18 ; 118 ; 18)) & */TextColor (If (IsEmpty (CCMabbrev) ; "" ; "; " & CCMabbrev) &If (IsEmpty (RISMabbrev) ; "" ; "; " & RISMabbrev) &If (IsEmpty (altRISMabbrev) ; "" ; "; " & altRISMabbrev) ; RGB (196 ; 5 ; 5)) ; Bold) ; Italic) ; "Arial") ; 13))*
Complete Library info_g	Text	Global

completeMS	Text	says whether the MS is complete or a bit of one
containerKey	Calculation	(Number) from Source, = "999"
dateComments	Text	Lookup: When an entry in "MOTET to source::sourceKey" is made, copy the first matching value from "Motetten-Datenbank::07. Date-Comments" If no match: "Do not copy" don't copy "Motetten-Datenbank::07. Date-Comments" if empty
dateOfSource	Text	
description	Calculation	(Text) Indexed, from Source, = *Substitute (If (IsEmpty (external_URLs) ; "" ; TextStyleAdd ("External data sources: " ; Bold) & external_URLs & ¶) &If (IsEmpty (description_DIAMM & description_RISM & description_CCM) ; "Noonline description for this MS is currently available." & ¶ & "If you would like to submit a description for this MS please send it todiamm[at]music.ox.ac.uk for consideration by the editorial board.Authorship will be acknowledged." & ¶;If (IsEmpty (description_DIAMM) ; "" ; TextStyleAdd ("DIAMM information(this information supersedes that given in RISM or CCM)" ; Bold) & If(IsEmpty (description_author) ; ":" ; " (" & description_author & "):") & ¶ &description_DIAMM & ¶) &If (IsEmpty (description_CCM) ;"" ; TextStyleAdd ("Census Catalogue Description:" ; Bold) & ¶ &"See Census-Catalogue of Manuscript Sources of Polyphonic Music 1400-1550 5 vols., Renaissance Manuscript Studies (Neuhausen-Stuttgart:AmericanInstitute of Musicology, Hänssler Verlag, 1979-1988)" & " Vol. " &Substitute (Substitute (Left (CCMimagefilename1_delete ; 4) ; "CCM" ; "")& ", p. " & Right (CCMimagefilename1_delete ; 3) &If (IsEmpty (CCMimagefilename2_delete) ; "" ; " and Vol. " & Substitute (Left(CCMimagefilename2_delete ; 4) ; "CCM" ; "") & ", p. " & Right(CCMimagefilename2_delete ; 3)) ; ["p. 00" ; "p. "] ; ["p. 0" ; "p. "]) & ¶& "This information is reproduced here by kind permission of the publishers. Itis COPYRIGHT and copying/reproduction of any of this content without permissionmay result in legal action." & ¶ & description_CCM & ¶)&If (IsEmpty (description_RISM) ; "" ; TextStyleAdd ("RISM description:" ; Bold) & ¶ & description_RISM & ¶)) &If (IsEmpty (inventory) ; "Please send corrections or new information todiamm[at]music.ox.ac.uk" ; TextStyleAdd ("TABLE OFCONTENTS" ; Bold) & ¶ & inventory & ¶ & "Please sendcorrections or new information to diamm[at]music.ox.ac.uk") ; "¶¶" ; "¶")*
description_author	Text	
description_CCM	Text	
description_DIAMM	Text	
description_RISM	Text	
done	Text	
editing_history	Calculation	(Text) Unstored, from Source, = *If (IsEmpty (logged_in_as) ; editing_history ; Get (AccountName) & " " & modification date & ¶ & editing_history)*
enddate	Text	
external_URLs	Text	
FlagCCM	Calculation	(Text) from Source, = *If (IsEmpty (CCMimagefilename1_delete) ; "" ; "CCM entry")*
FlagCCM 2	Calculation	(Text) from Source, = *If (IsEmpty (CCMimagefilename2_delete) ; "" ; "CCM second entry")*
FlagRISM	Calculation	(Text) from Source, = *If (IsEmpty (RISMimagefilename1) ; "" ; "RISM*

		entry")
FlagRISM 2	Calculation	(Text) from Source, = *If (IsEmpty (RISMimagefilename2) ; "" ; "RISM second entry")*
folio	Text	
format	Text	
g_bibliogAbbrev	Text	Global
g_Key	Calculation	(Number) from Source, = "99"
inventory	Text	
last edited by	Text	
letter_info	Calculation	(Text) from Source, = *If (done = "Y" ; If (webpermission = "Y" ; " (DIAMM has images of this MS online)" ; " (DIAMM has images of this MS)") ; "")*
logged_in_as	Calculation	(Text) from Source, = *Get (AccountName)*
modification date	Text	Auto-enter: "Modification Timestamp (Date and Time)"
MOTETshelfmark	Text	Lookup: When an entry in "MOTET to source::sourceKey" is made, copy the first matching value from "Motetten-Datenbank::MOTET shelfmark" If no match: "Do not copy" don't copy "Motetten-Datenbank::MOTET shelfmark" if empty
moved	Calculation	(Text) from Source, = *If (moved_Y = "Y"; "Please note: this document has been moved to another library!" ; "") INCORPORATE INTO DESCRIPTION?*
moved_Y	Text	
noofimages	Calculation	(Number) Unstored, from Source, = *Count (Image::source)*
noofimages_editable	Number	Lookup: When an entry in "Source::archiveKey" is made, copy the first matching value from "Source to ID::noofimages" If no match: "Do not copy" don't copy "Source to ID::noofimages" if empty
notation	Text	
notes	Text	
numberFound	Calculation	(Number) Unstored, from Source, = *Get (FoundCount)*
numbering_source	Text	
olim	Text	
pageMeasurements	Text	
pageNextCCM	Calculation	(Text) Unstored, from Source, = *Left (CCMimagefilename1_delete ; 5) & Case (Length (Right (CCMimagefilename1_delete ; 4) + 1) = 4 ; Right (CCMimagefilename1_delete ; 4) + 1 ; Length (Right (CCMimagefilename1_delete ; 3) + 1) = 3 ; Right (CCMimagefilename1_delete ; 3) + 1 ; Length (Right (CCMimagefilename1_delete ; 3) + 1) = 2 ; "0" & Right (CCMimagefilename1_delete ; 3) + 1 ; Length (Right (CCMimagefilename1_delete ; 3) + 1) = 1 ; "00" & Right (CCMimagefilename1_delete ; 3) + 1 ; "Whoops - you've run out of pages in this volume !")*
pageNextCCM_2	Calculation	(Text) Unstored, from Source, = *Left (CCMimagefilename2_delete ; 5) & Case (Length (Right (CCMimagefilename2_delete ; 4) + 1) = 4 ; Right (CCMimagefilename2_delete ; 4) + 1 ; Length (Right (CCMimagefilename2_delete ; 3) + 1) = 3 ; Right (CCMimagefilename2_delete ; 3) + 1 ; Length (Right (CCMimagefilename2_delete ; 3) + 1) = 2 ; "0" & Right (CCMimagefilename2_delete ; 3) + 1 ; Length (Right (CCMimagefilename2_delete ; 3) + 1) = 1 ; "00" & Right (CCMimagefilename2_delete ; 3) + 1 ; "Whoops - you've run out of pages in this volume !")*
pageNextRISM	Calculation	(Text) Unstored, from Source, = *Left (RISMimagefilename1 ; 2) & If (Length (RISMimagefilename1) = 5 ; Case (Length (Right (RISMimagefilename1 ; 3) + 1) = 3 ; Right (RISMimagefilename1 ; 3) + 1 ; Length (Right (RISMimagefilename1 ; 3) + 1) = 2 ; "0" & Right (RISMimagefilename1 ; 3) + 1 ; Length (Right (RISMimagefilename1 ; 3) +*

		1) = 1 ; "00" & Right (RISMimagefilename1 ; 3) + 1 ; "Whoops - you've run out of pages in this volume !") ; Case (Length (Right (RISMimagefilename1 ; 4) + 1) = 4 ; Right (RISMimagefilename1 ; 4) + 1 ; Length (Right (RISMimagefilename1 ; 4) + 1) = 3 ; "0" & Right (RISMimagefilename1 ; 4) + 1 ; Length (Right (RISMimagefilename1 ; 4) + 1) = 2 ; "00" & Right (RISMimagefilename1 ; 4) + 1 ; Length (Right (RISMimagefilename1 ; 4) + 1) = 1 ; "000" & Right (RISMimagefilename1 ; 4) + 1 ; "Whoops - you've run out of pages in this volume !"))
pageNextRISM 2	Calculation	(Text) Unstored, from Source, *= Left (RISMimagefilename2 ; 2) & If (Length (RISMimagefilename2) = 5 ; Case (Length (Right (RISMimagefilename1 ; 3) + 1) = 3 ; Right (RISMimagefilename2 ; 3) + 1 ; Length (Right (RISMimagefilename1 ; 3) + 1) = 2 ; "0" & Right (RISMimagefilename2 ; 3) + 1 ; Unstored, from Source, = Left (RISMimagefilename2 ; 2) & If (Length (RISMimagefilename2) = 5 ; Case (Length (Right (RISMimagefilename1 ; 3) + 1) = 3 ; Right (RISMimagefilename2 ; 3) + 1 ; Length (Right (RISMimagefilename1 ; 3) + 1) = 2 ; "0" & Right (RISMimagefilename2 ; 3) + 1 ; Length (Right (RISMimagefilename1 ; 3) + 1) = 1 ; "00" & Right (RISMimagefilename2 ; 3) + 1 ; "Whoops - you've run out of pages in this volume !") ; Case (Length (Right (RISMimagefilename1 ; 4) + 1) = 4 ; Right (RISMimagefilename2 ; 4) + 1 ; Length (Right (RISMimagefilename1 ; 4) + 1) = 3 ; "0" & Right (RISMimagefilename2 ; 4) + 1 ; Length (Right (RISMimagefilename1 ; 4) + 1) = 2 ; "00" & Right (RISMimagefilename2 ; 4) + 1 ; Length (Right (RISMimagefilename1 ; 4) + 1) = 1 ; "000" & Right (RISMimagefilename2 ; 4) + 1 ; "Whoops - you've run out of pages in this volume !"))*
pagePreviousCCM	Calculation	(Text) Unstored, from Source, *= Left (CCMimagefilename1_delete ; 5) & Case (Length (Right (CCMimagefilename1_delete ; 4) - 1) = 4 ; Right (CCMimagefilename1_delete ; 4) - 1 ; Length (Right (CCMimagefilename1_delete ; 3) - 1) = 3 ; Right (CCMimagefilename1_delete ; 3) - 1 ; Length (Right (CCMimagefilename1_delete ; 3) - 1) = 2 ; "0" & Right (CCMimagefilename1_delete ; 3) - 1 ; Length (Right (CCMimagefilename1_delete ; 3) - 1) = 1 ; "00" & Right (CCMimagefilename1_delete ; 3) - 1 ; "Whoops - you've run out of pages in this volume !")*
pagePreviousCCM_2	Calculation	(Text) Unstored, from Source, *= Left (CCMimagefilename2_delete ; 5) & Case (Length (Right (CCMimagefilename2_delete ; 4) - 1) = 4 ; Right (CCMimagefilename2_delete ; 4) - 1 ; Length (Right (CCMimagefilename2_delete ; 3) - 1) = 3 ; Right (CCMimagefilename2_delete ; 3) - 1 ; Length (Right (CCMimagefilename2_delete ; 3) - 1) = 2 ; "0" & Right (CCMimagefilename2_delete ; 3) - 1 ; Length (Right (CCMimagefilename2_delete ; 3) - 1) = 1 ; "00" & Right (CCMimagefilename2_delete ; 3) - 1 ; "Whoops - you've run out of pages in this volume !")*
pagePreviousRISM	Calculation	(Text) Unstored, from Source, *= Left (RISMimagefilename1 ; 2) & If (Length (RISMimagefilename1) = 5 ; Case (Length (Right (RISMimagefilename1 ; 3) - 1) = 3 ; Right (RISMimagefilename1 ; 3) - 1 ; Length (Right (RISMimagefilename1 ; 3) - 1) = 2 ; "0" & Right (RISMimagefilename1 ; 3) - 1 ; Length (Right (RISMimagefilename1 ; 3) - 1) = 1 ; "00" & Right (RISMimagefilename1 ; 3) - 1 ; "Whoops - you've run out of pages in this volume !") ; Case (Length (Right (RISMimagefilename1 ; 4) - 1) = 4 ; Right (RISMimagefilename1 ; 4) - 1 ; Length (Right (RISMimagefilename1 ; 4) - 1) = 3 ; "0" & Right (RISMimagefilename1 ; 4) - 1 ; Length (Right (RISMimagefilename1 ; 4) - 1) = 2 ; "00" & Right (RISMimagefilename1 ; 4) - 1 ; Length (Right (RISMimagefilename1 ; 4*

		) - 1) = 1 ; "000" & Right (RISMimagefilename1 ; 4) - 1 ; "Whoops - you've run out of pages in this volume !") Length (Right (RISMimagefilename1 ; 4) - 1) = 4 ; Right (RISMimagefilename1 ; 4) - 1 ; Length (Right (RISMimagefilename1 ; 4) - 1) = 3 ; "0" & Right (RISMimagefilename1 ; 4) - 1 ; Length (Right (RISMimagefilename1 ; 4) - 1) = 2 ; "00" & Right (RISMimagefilename1 ; 4) - 1 ; Length (Right (RISMimagefilename1 ; 4) - 1) = 1 ; "000" & Right (RISMimagefilename1 ; 4) - 1 ; "Whoops - you've run out of pages in this volume !"))
pagePreviousRISM 2	Calculation	(Text) Unstored, from Source, = *Left (RISMimagefilename2 ; 2) & If (Length (RISMimagefilename2) = 5 ; Case (Length (Right (RISMimagefilename1 ; 3) - 1) = 3 ; Right (RISMimagefilename2 ; 3) - 1 ; Length (Right (RISMimagefilename1 ; 3) - 1) = 2 ; "0" & Right (RISMimagefilename2 ; 3) - 1 ; Length (Right (RISMimagefilename1 ; 3) - 1) = 1 ; "00" & Right (RISMimagefilename2 ; 3) - 1 ; "Whoops - you've run out of pages in this volume !") ; Case (Length (Right (RISMimagefilename1 ; 4) - 1) = 4 ; Right (RISMimagefilename2 ; 4) - 1 ; Length (Right (RISMimagefilename1 ; 4) - 1) = 3 ; "0" & Right (RISMimagefilename2 ; 4) - 1 ; Length (Right (RISMimagefilename1 ; 4) - 1) = 2 ; "00" & Right (RISMimagefilename2 ; 4) - 1 ; Length (Right (RISMimagefilename1 ; 4) - 1) = 1 ; "000" & Right (RISMimagefilename2 ; 4) - 1 ; "Whoops - you've run out of pages in this volume !"))*
Phase	Text	
provenance	Text	
RISMabbrev	Text	
RISMCCM	Text	
RISMimagefilename1	Text	
RISMimagefilename2	Text	
RISMimagefilename3	Text	
RISMpresence	Calculation	(Text) from Source, = *If (IsEmpty (RISMimagefilename1) ; "" ; "RISM")*
RISMreset	Text Auto-enter	Calculation: from Source, = RISMimagefilename1
setKey	Number	
shelfMark	Calculation	(Text) from Source, = *shelfMark alone* & " *[*" & *Tags* & "*]*"
shelfMark_tagged	Text	
sortorder	Number	Allow user to override validation, Required value
sourceDisplayName	Calculation	(Text) from Source, = *shelfMark alone & If (IsEmpty (sourceName) ; "" ; ", " & "\"" & sourceName & "\"") & If (IsEmpty (olim) ; "" ; ", olim: " & olim)*
sourceKey	Number	Serial Number on creation with Current Value: "3863" Increment: "1", Required value, Unique values only, Only allow values of type: "Numeric Only"
sourceName	Text	Lookup: When an entry in "MOTET to source::sourceKey" is made, copy the first matching value from "Motetten-Datenbank::01. SourceName" If no match: "Do not copy" don't copy "Motetten-Datenbank::01. SourceName" if empty
sourcesummary	Calculation	(Text) from Source, = *shelfMark alone* & " *fols.* " & *folio*
sourceType	Text	
startdate	Text	
status	Text	
staveGauge	Text	
summary	Number	Auto-enter Calculation: from Source, = Sum (noofimages)
summaryfield	Calculation	(Text) Unstored, from Source, = *TextFont (TextColor (city* & " " & *archive* & " " & *shelfMark alone* & " " & *folio & If (IsEmpty (noofimages) ; "" ; " (total images:* " & *noofimages* & ")") ; *RGB (255 ; 255 ; 255)) ; "Arial")*

surface	Text	
Tags	Calculation (Text)	from Source, = *If (useDescription_for_tag = "1" ; description_DIAMM ; If (IsEmpty (dateOfSource) ; "" ; dateOfSource & " ") & If (IsEmpty (sourceType) ; "" ; sourceType) & "; " & If (IsEmpty (provenance) ; "provenance unknown" ; "provenance: " & provenance) & If (IsEmpty (sourceName) ; "" ; ", " & sourceName))*
useDescription_for_tag	Text	
webpermission	Text	Auto-enter: "N"

Source_Copyist_IS

Intersection between the **Source** *and* **alCopyist** *tables. The type of copyist (e.g.. music, text, illuminator etc.) is defined in* **alCopyistType**, *which is connected to these tables via this intersection set, since a copyist may appear in various books but not always in the same role, so the information has to be contained in the intersection*

Field Name	Field Type	Formula / Entry Option
attribution_uncertain	Text	
copyistKey	Number	
copyistTypeKey	Number	
source_Copyist_IS_KEY	Number	Serial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
sourceKey	Number	

Source_Person_IS

intersection between **Source** *and* **alPerson**, *allowing more than one person to be linked to any source, and more than one source to be linked to a person.*

Field Name	Field Type	Formula / Entry Option
attribution_uncertain	Text	
personKey	Number	
personRelationshipKey	Number	
sourceKey	Number	
source_Person_IS_KEY	Number	Serial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"

SourceProvenance_IS

Intersection between the **Source** *and* **alProvenance** *tables, implemented in order to limit the Provenance authority list to countries only: the intersection set contains more detailed information about provenance where that is known, such as region, protectorate, city and institution.*

Field Name	Field Type	Formula / Entry Option
city	Text	
institution	Text	
protectorate	Text	
provenanceKey	Number	
region	Text	
sourceKey	Number	
sourceProvenance_IS_KEY		NumberSerial Number on creation with Current Value: "3008" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"
uncertain	Text	

SourceSet_IS

The intersection set between the **Source** *and* **Set** *tables, allowing sources to belong to more than one set.*

Field Name	Field Type	Formula / Entry Option
setKey	Number	
sourceKey	Number	
sourceSet_IS_KEY	Number	Serial Number on creation with Current Value: "1065" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"

Text

This table contains the text records linked to items in the **Item** *table. The records include primarily text incipits in original and standardised spelling, but also contain a number of complete text transcriptions. Each record represents one voice part. Where a voice part exists but is untexted it is nevertheless created in this table as this allows a voice part to be linked to clefs, mensurations and voice designations, all 'satellite' tables to this one. There is also a further satellite authority list,* **alLanguage**, *which allows us to define the language or languages used in a particular voice part via an intersection set:* **TextLanguage_IS**.

Field Name	Field Type	Formula / Entry Option
alGenreKey	Number	Lookup: When an entry in "Text::itemKey" is made, copy the first matching value from "Item::genreKey" If no match: "Do not copy" don't copy "Item::genreKey" if empty
canon	Text	
clef	Text	
clefKey	Number	
completeLibraryInfo	Text	Lookup: When an entry in "Text::itemKey" is made, copy the first matching value from "Item to Source::Complete Library info" If no match: "Do not copy" don't copy "Item to Source::Complete Library info" if empty
editorial_voicepart	Text	
folios	Text	Lookup: When an entry in "Text::itemKey" is made, copy the first matching value from "Item::Folios" If no match: "Do not copy" don't copy "Item::Folios" if empty
fullItemText	Text	
fullItemText_authority	Text	
g_voiceKey	Calculation	(Number) from Text, = "999"
Genre	Text	Lookup: When an entry in "Text::itemKey" is made, copy the first matching value from "alGenre::Genre" If no match: "Do not copy" don't copy "alGenre::Genre" if empty
Global	Number	Auto-enter Calculation: from Text, = "999" Global field used in relationship between Item and Text to show all records
imageKey	Text	Lookup: When an entry in "Text::incipitfilename" is made, copy the first matching value from "Text to ItemImage::imageKey" If no match: "Do not copy" don't copy "Text to ItemImage::imageKey" if empty
incipitfilename	Text	Lookup: When an entry in "Text::itemKey" is made, copy the first matching value from "Item::incipitfilename" If no match: "Do not copy" don't copy "Item::incipitfilename" if empty
itemKey	Number	
mensurationKey	Number	
orderNo	Number	Serial Number on creation with Current Value: "27237" Increment: "1"
positiononPage	Text	Lookup: When an entry in "Text::incipitfilename" is made, copy the first matching value from "Text to ItemImage::positiononPage" If no match: "Do not copy" don't copy "Text to ItemImage::positiononPage" if empty
sourceKey	Number	Lookup: When an entry in "Text::itemKey" is made, copy the first matching value from "Item::sourceKey" If no match: "Do not copy" don't copy "Item::sourceKey" if empty
standardspellingFulltext	Text	
standardspellingIncipit	Text	
textincipit	Text	

TextKey	Number	Serial Number on creation with Current Value: "27278" Increment: "1", Required value, Unique values only, Only allow values of type: "Numeric Only"
voiceKey	Number	
voicepart	Text	

TextLanguage_IS

Intersection set between **Text** *and* **Language** *allowing the text of a voice part to be linked to more than one language (e.g. a latin sacred text that has a vernacular trope).*

Field Name	Field Type	Formula / Entry Option
languageKey	Number	
textKey	Number	
textLanguage_IS_KEY	Number	Serial Number on creation with Current Value: "601" Increment: "1", Allow user to override validation, Required value, Unique values only, Only allow values of type: "Numeric Only"

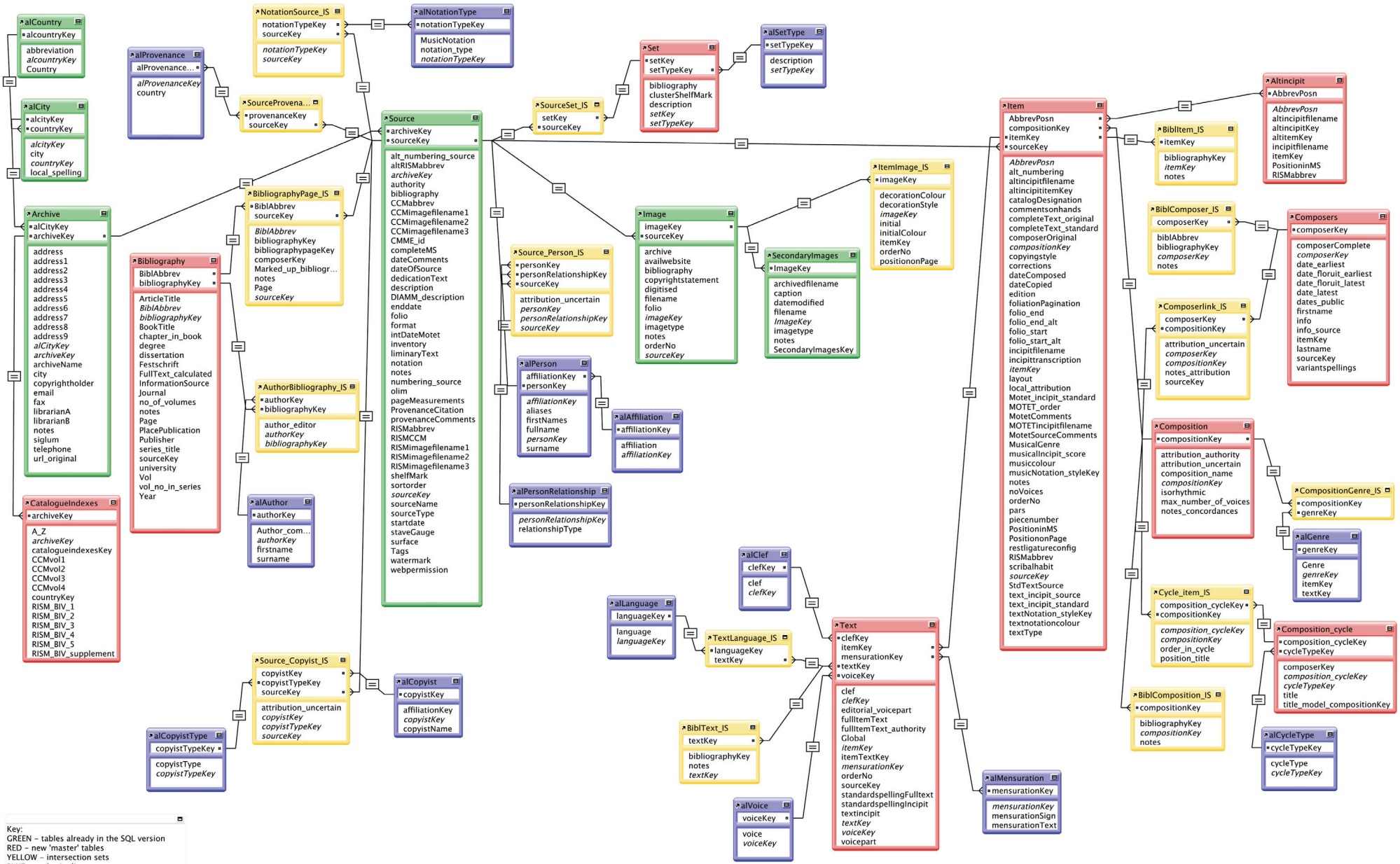
db_globals

In order to facilitate navigation around the database for editors this table and its fields have been added to operate 'back' and 'forward' buttons placed in layout headers. The scripting and calculations of this table are not reproduced here as they are the intellectual property of Robin Ledward.

2.4 Entity Relationship Diagram (ERD)

The historical development of the database has been described in section 1, above, together with various versions of the Entity Relationship Diagram. The final ERD at the cessation of development and content input work in May 2011 is represented by the database table and field descriptions above, and the complete ERD given below. This ERD represents the idealised database structure. The structure of the working database includes numerous links and relationships not shown here that are used solely to facilitate the construction of working layouts that create specific environments for different types of data input, such as the inclusion of portal listings of content from other related tables shown in a master-table layout.

These redundancies have not been removed because they all perform some function in assisting non-expert users to populate, read and use the Filemaker database. It might be useful at some future time to reconstruct the database from scratch to try and eliminate some of these additional structures, as well as some of the layouts that were designed for particular tasks, now completed, but they have been left in place for the time being.


alCountry
alcountryKey
abbreviation
alcountryKey
Country
alCity
alcityKey
countryKey
alcityKey
city
countryKey
local_spelling
Archive
alCityKey
archiveKey
address
address1
address2
address3
address4
address5
address6
address7
address8
address9
alCityKey
archiveKey
archiveName
city
copyrightholder
email
fax
librarianA
librarianB
notes
siglum
telephone
url_original
CatalogueIndexes
archiveKey
A_Z
archiveKey
catalogueindexesKey
CCMvol1
CCMvol2
CCMvol3
CCMvol4
countryKey
RISM_BIV_1
RISM_BIV_2
RISM_BIV_3
RISM_BIV_4
RISM_BIV_5
RISM_BIV_supplement
alProvenance
alProvenance…
alProvenanceKey
country
NotationSource_IS
notationTypeKey
sourceKey
notationTypeKey
sourceKey
alNotationType
notationTypeKey
MusicNotation
notation_type
notationTypeKey
SourceProvena…
provenanceKey
sourceKey
Source
archiveKey
sourceKey
alt_numbering_source
altRISMabbrev
archiveKey
authority
bibliography
CCMabbrev
CCMimagefilename1
CCMimagefilename2
CCMimagefilename3
CMME_id
completeMS
dateComments
dateOfSource
dedicationText
description
DIAMM_description
enddate
folio
format
intDateMotet
inventory
liminaryText
notation
notes
numbering_source
olim
pageMeasurements
ProvenanceCitation
provenanceComments
RISMabbrev
RISMCCM
RISMimagefilename1
RISMimagefilename2
RISMimagefilename3
shelfMark
sortorder
sourceKey
sourceName
sourceType
startdate
staveGauge
surface
Tags
watermark
webpermission
BibliographyPage_IS
BiblAbbrev
sourceKey
BiblAbbrev
bibliographyKey
bibliographypageKey
composerKey
Marked_up_bibliogr…
notes
Page
sourceKey
Bibliography
BiblAbbrev
bibliographyKey
ArticleTitle
BiblAbbrev
bibliographyKey
BookTitle
chapter_in_book
degree
dissertation
Festschrift
FullText_calculated
InformationSource
Journal
no_of_volumes
notes
Page
PlacePublication
Publisher
series_title
sourceKey
university
Vol
vol_no_in_series
Year
AuthorBibliography_IS
authorKey
bibliographyKey
author_editor
authorKey
bibliographyKey
alAuthor
authorKey
Author_com…
authorKey
firstname
surname
SourceSet_IS
setKey
sourceKey
Set
setKey
setTypeKey
bibliography
clusterShelfMark
description
setKey
setTypeKey
alSetType
setTypeKey
description
setTypeKey
Source_Person_IS
personKey
personRelationshipKey
sourceKey
attribution_uncertain
personKey
personRelationshipKey
sourceKey
alPerson
affiliationKey
personKey
affiliationKey
aliases
firstNames
fullname
personKey
surname
alAffiliation
affiliationKey
affiliation
affiliationKey
alPersonRelationship
personRelationshipKey
personRelationshipKey
relationshipType
Image
imageKey
sourceKey
archive
availwebsite
bibliography
copyrightstatement
digitised
filename
folio
imageKey
imagetype
notes
orderNo
sourceKey
ItemImage_IS
imageKey
decorationColour
decorationStyle
imageKey
initial
initialColour
itemKey
orderNo
positiononPage
SecondaryImages
ImageKey
archivedfilename
caption
datemodified
filename
ImageKey
imagetype
notes
SecondaryImagesKey
Item
AbbrevPosn
compositionKey
itemKey
sourceKey
AbbrevPosn
alt_numbering
altincipitfilename
altincipititemKey
catalogDesignation
commentsonhands
completeText_original
completeText_standard
composerOriginal
compositionKey
copyingstyle
corrections
dateComposed
dateCopied
edition
foliationPagination
folio_end
folio_end_alt
folio_start
folio_start_alt
incipitfilename
incipittranscription
itemKey
layout
local_attribution
Motet_incipit_standard
MOTET_order
MotetComments
MOTETincipitfilename
MotetSourceComments
MusicalGenre
musicalIncipit_score
musiccolour
musicNotation_styleKey
notes
noVoices
orderNo
pars
piecenumber
PositioninMS
PositiononPage
restligatureconfig
RISMabbrev
scribalhabit
sourceKey
StdTextSource
text_incipit_source
text_incipit_standard
textNotation_styleKey
textnotationcolour
textType
Altincipit
AbbrevPosn
AbbrevPosn
altincipitfilename
altincipitKey
altitemKey
incipitfilename
itemKey
PositioninMS
RISMabbrev
BiblItem_IS
itemKey
bibliographyKey
itemKey
notes
BiblComposer_IS
composerKey
biblAbbrev
bibliographyKey
composerKey
notes
Composers
composerKey
composerComplete
composerKey
date_earliest
date_floruit_earliest
date_floruit_latest
date_latest
dates_public
firstname
info
info_source
itemKey
lastname
sourceKey
variantspellings
Composerlink_IS
composerKey
compositionKey
attribution_uncertain
composerKey
compositionKey
notes_attribution
sourceKey
Composition
compositionKey
attribution_authority
attribution_uncertain
composition_name
compositionKey
isorhythmic
max_number_of_voices
notes_concordances
CompositionGenre_IS
compositionKey
genreKey
alGenre
genreKey
Genre
genreKey
itemKey
textKey
Cycle_item_IS
composition_cycleKey
compositionKey
composition_cycleKey
compositionKey
order_in_cycle
position_title
Composition_cycle
composition_cycleKey
cycleTypeKey
composerKey
composition_cycleKey
cycleTypeKey
title
title_model_compositionKey
BiblComposition_IS
compositionKey
bibliographyKey
compositionKey
notes
alCycleType
cycleTypeKey
cycleType
cycleTypeKey
alClef
clefKey
clef
clefKey
alLanguage
languageKey
language
languageKey
TextLanguage_IS
languageKey
textKey
Text
clefKey
itemKey
mensurationKey
textKey
voiceKey
clef
clefKey
editorial_voicepart
fullItemText
fullItemText_authority
Global
itemKey
itemTextKey
mensurationKey
orderNo
sourceKey
standardspellingFulltext
standardspellingIncipit
textincipit
textKey
voiceKey
voicepart
BiblText_IS
textKey
bibliographyKey
notes
textKey
alVoice
voiceKey
voice
voiceKey
alMensuration
mensurationKey
mensurationKey
mensurationSign
mensurationText
Source_Copyist_IS
copyistKey
copyistTypeKey
sourceKey
attribution_uncertain
copyistKey
copyistTypeKey
sourceKey
alCopyist
copyistKey
affiliationKey
copyistKey
copyistName
alCopyistType
copyistTypeKey
copyistType
copyistTypeKey
Key:
GREEN – tables already in the SQL version
RED – new 'master' tables
YELLOW – intersection sets
BLUE – authority lists
BLACK – KCL tables

3 USER INFORMATION

Introduction

An understanding of the way a relational database works is essential to understanding how data is managed within this database. The foregoing documentation should explain this sufficiently. Working with a database is VERY DIFFERENT from working with work-processed files: if you want to add a bibliography item to a source description you cannot simply cut-and-paste it into the source description. The bibliographical item has to be created in the bibliography database (assuming it is not there already, so that has to be checked too), marked up by distributing the information into individual fields (title, authors, journal, date etc.); that record has to be linked to the Author database, and only then can the bibliography record be linked to the source you wanted it to link to.

In the same way, deleting material is not simply a question of hitting 'delete'. If you delete a source record, you must first delete all dependent records - items, images, bibliography items etc. that are connected to the source record. You cannot delete items without also first deleting the text, clef, language, voice-designation etc. dependencies to each of those records, and in many cases this means also chasing down bibliographical references to item records and deleting those dependencies as well, before you can delete the item.

Records are connected by unique key numbers, but if you inadvertently change a key number you can completely disconnect and corrupt the data that flows from it. For example if, in editing a Source record, you accidentally delete or change the ArchiveKey number, that manuscript may end up linked to the wrong library, or linked to no library at all, so it will no longer appear where it should, nor will any of its dependencies (items, text transcriptions, bibliography, images etc.). Key numbers can however be used to make quick changes very easily: a new source record is placed in the correct library (and city and country) simply by inserting the relevant archiveKey number into the Source record.

Warning: DO NOT at any time attempt to change the Key number of a record. Just don't.

3.1 Editing existing records

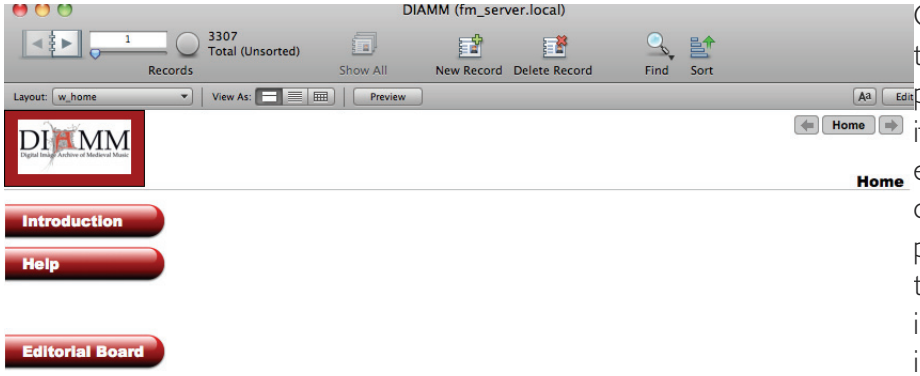
This is relatively simple and only really involves finding the right record and editing the relevant field(s). The database has an entry page which will gives users less experienced with databases a step-by-step process to find the records they want to edit and work on them without requiring an intimate understanding of the workings of the database. The layouts can be used both in web publishing format to be accessed through a normal web-broser and for users accessing the database through Filemaker software, but only those designed for the editorial board are optimized for web use.

Although Filemaker is cross-platform there are issues with variable font sizes between Mac and PC which mean that some fonts appear larger on the PC than they do on a Mac, making field heights look wrong in some layouts. Since adjusting every field in every layout would be an inefficient use of time, we ask PC-users to be tolerant of these variations. A web browser will also change some fonts (despite the use of web-compatible fonts throughout), but this has not created any immediate problems.

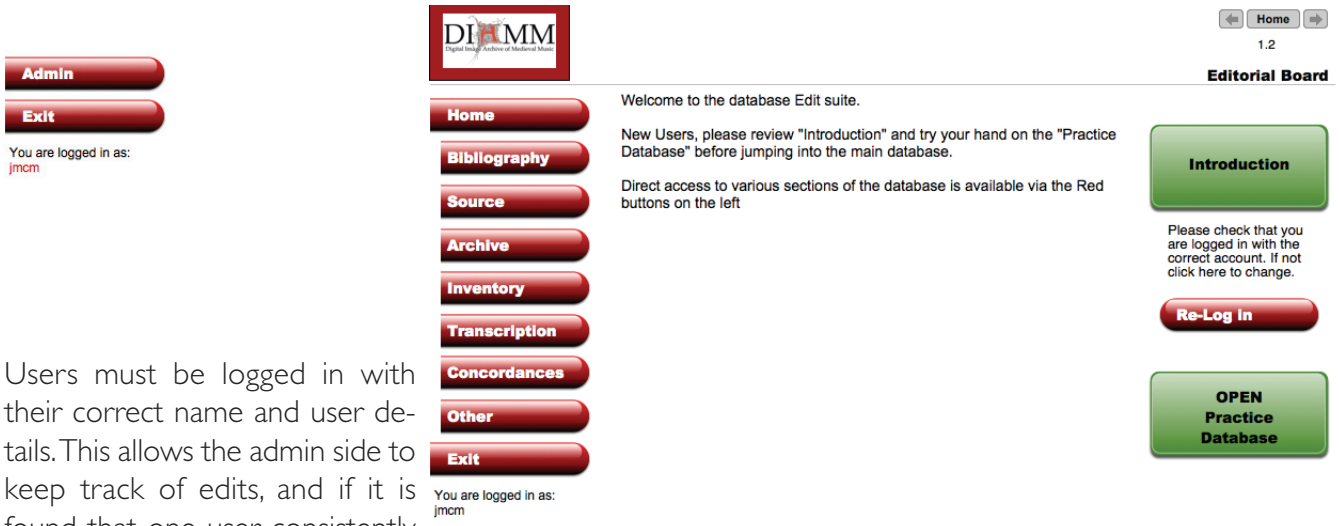
One caveat regarding versions: FM v.9 will run most scripts in the database, and should run all the scripts. However we have found that many of the editorial board layouts require FM v.10 to run the scripts in that section (which are more complex in order to simply access to the end user) correctly. The main visual difference between v.9 and v.10 is that the toolbar in v.9 appears on the left edge of the view, and has been moved to the top of the view in v.10. This is something of an annoyance in layouts that were optimised for vertical screen real-estate, since they now appear chopped off at the bottom edge.

This section of the database documentation does not provide a user manual as such, since the layouts accessed from the entry layout (or database 'Home' page) are documented with instructions for use *on the layout* so that instructions for each step are available at the point of access. The Editorial Board should access the

database from the 'Editorial Board' button, and more experienced users from the 'Admin' button.



Clicking the Editorial Board button takes users into the heavily scripted part of the database where all activity is carefully guided. from a series of explicit buttons. At any point the user can return to this master layout. A practice database (basically a copy of the master database) is available for inexperienced users to try out editing processes without worrying that they are affecting the master data.

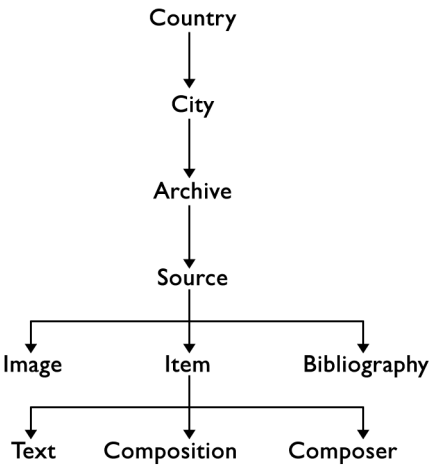


Users must be logged in with their correct name and user details. This allows the admin side to keep track of edits, and if it is found that one user consistently makes a particular mistake, their activity can be tracked and corrected. Similarly, if a problem arises in the data it is easy to see with which user it originated. Finally, if the database crashes following an edit, and the user was not able to record the record ID where they were working, the audit trail saved by the login information enables an administrator to find the problem and correct it without record ID numbers.

Editing is based around the four main tables in the database: Archive, Source, Item and Bibliography. Content from other related tables is, for the most part, visible from the master layouts for these sources, enabling users to edit content without constantly changing layout or opening new windows. Global Key fields in each table allow complete table content lists to be shown on any related layout, as well as lists of data related specifically to the record being viewed/edited.

3.2 Creating new records

Creating records in a relational database is more complex than editing because the user must not attempt to create a record until the authority lists preceding it have been populated correctly. Thus if a new MS Source is to be added to the database the user must first check that the required Archive is already in the database, and if not that must be created. Likewise, a new Archive cannot be created unless its Country and City are already in those authority lists. The BASIC hierarchy map shown here should therefore always be kept in mind when creating new records in the database: before creating any of the information shown in the lower subject areas, the information above it must be in place.



All of these processes are guided in the Editorial Board layouts in the database. If using the database it is important to read through all the instructions and make sure that they are fully understood before proceeding. If in doubt, the practice database can be used, but it is essential to remember that nothing done in the practice database will be saved or will appear online, so extensive editing will be lost.

Creating a new Archive:

Check to see that the country and city for your archive already exist before attempting to create a new archive. There is no point in creating archives unless they contain sources, so do not create archive records just because the library exists in the real world: the dataset ONLY includes archives in which there are manuscripts relevant to this dataset.

Creating a new Source:

Check to see that the Archive you want to put your source in already exists before you create the new Source record. You will need the archiveKey number in order to link the new manuscript record to that library. Do not add printed sources to this dataset unless they are unique, or contain manuscript additions.

Creating new Bibliographical citations:

Check first that the citation does not already exist - be careful to ensure that you do a sufficiently broad search that you will find any possible matches.

Creating new Inventory Items:

Inventories will be sorted by the order number of the items, so each piece in an inventory needs to be numbered. The foliation is not sufficient, since foliations vary so much that there is no consistent sort order that can be applied.

Julia Craig-McFeely, Oxford, April 2011